

Министерство просвещения РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологий  
Кафедра информатики, информационных технологий  
и методики обучения информатике

# РАЗРАБОТКА РХЕ СЕРВЕРА С УДАЛЕННЫМ ДОСТУПОМ

*Выпускная квалификационная работа  
бакалавра по направлению подготовки  
09.03.03 – Прикладная информатика в сервисе*

Исполнитель: студент группы ПИ 1601z  
Института математики, физики,  
информатики и технологий  
Сливканич Д.Д.

Работа допущена к защите  
«\_\_\_\_\_» \_\_\_\_\_ 2021 г.

Зав. кафедрой \_\_\_\_\_  
\_\_\_\_\_

Руководитель ОПОП \_\_\_\_\_  
\_\_\_\_\_

Руководитель: кандидат педагогических наук,  
доцент кафедры ИИТ и МОИ  
Рожина И.В.

Екатеринбург – 2021

## Реферат

Сливканич Д.Д. РАЗРАБОТКА PXE СЕРВЕРА С УДАЛЕННЫМ ДОСТУПОМ, выпускная квалификационная работа: 65стр., рис. 41, библи. назв. 20.

Ключевые слова: СЕРВЕР, PXE, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, УДАЛЕННЫЙ ДОСТУП, УСТАНОВКА, ДИАГНОСТИКА, DHCP, TFTP, HTTP, ОПЕРАЦИОННЫЕ СИСТЕМЫ, ИНЖЕНЕРЫ, НАКОПИТЕЛИ, ОБОРУДОВАНИЕ, ИНТЕРФЕЙС, МЕНЮ.

**Предметом разработки** является сервер PXE оснащенный необходимым набором программного обеспечения для диагностики серверного оборудования с возможностью удаленного управления для конфигурации и мониторинга состояния.

**Целью разработки** является разработка сервера PXE для диагностики серверного оборудования

В данной работе описаны результаты создания сервера PXE с удаленным доступом для технического отдела компании ООО НАГ.

Разработанный сервер создавался для загрузки рабочей станции через PXE были организованы взаимодействия служб DHCP, HTTP и TFTP с сервером PXE.

Разработанный сервер создавался на базе одноплатного компьютера ORANGE PI PC+, под управлением операционной системы ARMBIAN, для компьютеров с процессорами ARM, построенной на базе Linux Debian.

Данный сервер является актуальной и его использование значительно упрощает настройку оборудования и дальнейшее администрирование не только инженеров технического отдела данной компании, но и любого системного администратора.

# Оглавление

Введение .....	4
Глава 1. Теоретико-аналитическая часть .....	6
1.1 Обоснование постановки задачи .....	6
1.2 Выбор метода разработки .....	12
1.3 Формализованное описание технического задания.....	23
Глава 2. Практическая часть.....	26
2.1 Модельные представления объекта разработки .....	26
2.2 Описание продукта и демонстрация .....	47
2.3 Результаты апробации и внедрения. ....	59
Заключение .....	61
Список информационных источников.....	62
Приложения.....	64

## Введение

В современном научно-техническом, и быстро прогрессирующем мире, информационные технологии повсеместно внедряются в любые сферы деятельности. На учебе, на работе, дома, - сфера использования компьютеров огромна и безгранична, она постоянно растет, безусловно влияя на жизнь всего общества и развитие его умений, знаний и производственных сил. С развитием общества, науки развивается и ПЭВМ, совершенствуются и улучшаются технические характеристики оборудования, основными из которых являются: быстродействие, удобство в работе, стоимость, компактность и размеры, количество потребляемой электроэнергии.

До 90% IT-инфраструктуры предприятий и учреждений во всем мире, занимает оборудование, которое снято с производства вендором. Регулярно возникают ситуации, когда после апгрейда или реализации проектов интеграции, остаются невостребованные предыдущие модели или неиспользуемые остатки оборудования.

Невостребованное оборудование не выгодно хранить. В США и Европе практика возврата вложенных инвестиций давно стала популярной. В связи с этим увеличиваются как поставки данного оборудования, так и продажа его у нас в России. Надежность оборудования популярных брендов известна. Принципиально все отличие работоспособного used-оборудования от нового состоит только в цене, иногда в наличии фирменной упаковки и крепежа, и, возможно, в нескольких царапинах на корпусе. В данное время выросла популярность на данное оборудование, так-как его покупка очень выгодна, брендовое оборудование по очень хорошим ценам, которое почти не уступает по производительности новому оборудованию. Все поставляемое оборудование находится в отличном рабочем состоянии, и перед продажей проходит обязательное тестирование.

Для тестирования данного оборудования приходится пользоваться различным программным обеспечением для проверки его на работоспособность.

В связи с этим, иметь много Flash-накопителей не удобно, а еще и бессмысленная трата бюджета.

Для того, чтобы отказаться от множества накопителей и возможность одновременно производить диагностику сразу нескольких рабочих станций, мы можем использовать сервер PXE.

**Предметом разработки** является сервер PXE оснащенный необходимым набором программного обеспечения для диагностики серверного оборудования с возможностью удаленного управления для конфигурации и мониторинга состояния.

**Целью разработки** является разработка сервера PXE для диагностики серверного оборудования.

Задачи проекта:

- 1) Произвести анализ состояния проблемы и подходов к ее решению.
- 2) Произвести анализ и обосновать выбор используемой технологии.
- 3) В соответствии с техническим заданием провести разработку сервера PXE для диагностики оборудования.

# Глава 1. Теоретико-аналитическая часть

## 1.1 Обоснование постановки задачи

Компания, для которой будет разработан PXE сервер, торгует не только новым оборудованием, но и оборудованием, которое раньше уже подвергалось эксплуатации, но по результату предпродажной подготовки, компания гарантирует бесперебойную работу изделий, а также оказывает сервисную поддержку.

Предпродажная подготовка.

Данную подготовку проходят серверы и их компоненты, бывшие в употреблении.

Предпродажная подготовка оборудования, подразумевает под собой проверку изделия на наличие внешних дефектов, очистку от пыли, появившуюся в результате хранения оборудования на складе без упаковки. После внешней проверки оборудование проходит проверку внутренних компонентов и, если, дефектов не обнаружено, к оборудованию подключается питание для прохождения программных тестов.

В зависимости от сервера используется тот или иной инструмент:

- отвертки PH, SL, TORX, HEXAGONE;
- шарнирно-губцевые приспособления плоскогубцы, круглогубцы, бокорезы (боковые кусачки);
- канцелярский нож;
- USB съемные носители с записанным на них образами утилит и LIVE-образов;
- серверное оборудование для проверки работоспособности серверов;
- термопаста;
- жесткие диски для проверки дисковых полок, дисковых корзин серверов.

Благодаря сервисной поддержке и подготовки изделий к продаже, по средствам различных тестов, можно used-оборудование считать, ничем не хуже нового.

Производится продувка оборудования от пыли в специальной камере с компрессором, очистка корпуса от разнообразных инородных наклеек, очистка старой термопасты с процессора и нанесение новой, влажными и сухими салфетками протираются мелкие элементы сервера.

Производится проверка всех кабелей и комплектующих внутри сервера, проверка всех разъёмов под периферийные устройства.

Есть вероятность того, что оборудование долго хранилось на складе и некоторые контакты могли окислиться, на пример у таких компонентов как:

- оперативной памяти;
- памяти контроллера и самого контроллера;
- сетевых карт.

Их следует почистить, сделать это можно обычным ластиком.

Прежде чем приступить к тестированию оборудования, необходимо ознакомиться со всей документацией, предоставленной на оборудование, в случае отсутствия – запросить информацию у производителя или найти в интернете соответствующие руководства по эксплуатации данных изделий.

Список необходимого оборудования, которым укомплектовано каждое рабочее место инженеров по направлению “Серверы и СХД”:

- 1) Компьютерная мышь с интерфейсом usb;
- 2) Клавиатура с интерфейсом usb;
- 3) Монитор с интерфейсом VGA;
- 4) Flash накопители со специальными утилитами для проверки оборудования;
- 5) Кабели питания;
- 6) Патч-корд;
- 7) Ноутбук;

Каждая предпродажная подготовка подразумевает под собой совокупность стрессовых тестов, для того что бы клиенту отправить гарантированно рабочее оборудование, способное выдержать высокие нагрузки.

Проведение нагрузочного тестирования подразумевает следующую методику тестирования:

1) Подготовка – производится анализ тестируемого оборудования. Создается тактика для испытаний, направленных на определенные компоненты системы, перечень и описание тестов нагрузочного тестирования;

2) Тестирование – выполняется операция запуска тестов. Собирается статистическая информация, при помощи которой выполняется анализ производительности;

3) Результаты – после проведения всех видов тестирования формируется отчет, в котором предоставляются результаты тестирования.

Все виды испытаний информационной системы, которая соответствует совокупности программных и аппаратных средств разделяются на два основных вида:

- нефункциональное тестирование – призвано подтвердить или опровергнуть соответствие различных свойств надежности и производительности;
- функциональное тестирование – подтверждение то, что система предоставляет максимальную функциональность в данных условиях.

В данном случае необходимо разобрать более подробно функциональное тестирование, так как нагрузочное тестирование является подвидом функционального тестирования.

Подвиды функционального тестирования:

- нагрузочное тестирование – подразумевается тестирование системы, в режиме ожидаемой нагрузки, не превышающей возможности системы;
- стрессовое тестирование – заключается в испытание системы в условиях минимальных аппаратных ресурсов и максимально допустимых нагрузок;
- объемное тестирование – тестирование в условиях максимально допустимых объемов информации, позволяющая проверить зависимость времени отклика и производительности системы от объемов обрабатываемых данных;
- тестирование стабильности – проверка, заключающаяся в средней нагрузке на систему в течении продолжительного времени;



- тестирование надежности – объединение вышеупомянутых подвидов тестирования, функционально заключающиеся в проверке возможности системы возвращаться к исходному состоянию после прохождения всех до этого упомянутых тестов;

- тестирование эргономики – испытание интерфейса системы на безопасность и удобство работы с ним.

Штатные системы тестирования бывают разные, как в плане состава пакетов тестов, так и в плане пригодности для тестирования в данной сфере. Основным критерий для выбора систем тестирования – сохранение логов тестирования в текстовом варианте.

Тесты могут быть:

- предложены производителем, но от сторонних разработчиков;
- разработаны самим производителем данного оборудования;
- разработаны и встроены в сервер, как средство самодиагностики;
- от стороннего производителя, но с заявленной поддержкой данного оборудования.

Видов тестирования бывает очень много, но каждый из них имеет свои плюсы и недостатки.

Этапы диагностики сервера.

- 1) Визуальный осмотр на наличие дефектов корпуса.
- 2) Извлечение блоков питания, сравнение их партийных номеров, вольтажа, если блоки съемные (с технологией горячей замены).

Сервер оснащен множеством разнообразных датчиков и разный вольтаж блоков питания послужит причиной ошибки при самодиагностике сервера в момент включения. С несовпадением парт номеров, ситуация аналогична, разные парт-номера означают то, что блоки питания производились не в одной партии, а значит собраны из разных элементов. В процессе эксплуатации два блока с разными элементами, возможно будут распределять нагрузку сети по-разному, что может привести к выходу сервера из строя.

Есть серверы с одним не съёмным блоком питания, то есть блок питания встроен в сам корпус. В таком случае проверка блока питания не требуется, на наличие дефектов, он проверен при осмотре корпуса.

3) Подключение питания. Если, не были допущены ошибки, в предыдущем шаге диагностики сервера, он включится без инцидентов. Однако заметить поломку в блоках питания зачастую не представляется возможным, это связано с особенностью их не разборного корпуса. При поломке блока питания произойдет замыкания, сервер не пострадает, но необходимо незамедлительно отключить источник питания во избежание возгорания или удара током.

4) На уже запущенном сервере, происходит проверка на работоспособность интерфейсов:

- VGA;
- USB;
- PS/2;
- SATA/SAS;
- Ethernet;
- проверка редких дополнительных интерфейсов, вроде iSCSI, SFP, COM и т.д.

5) С запуском, производится сброс конфигурации оборудования к заводским настройкам, программно (в BIOS), либо аппаратно, по средствам извлечения батарейки CMOS, если программный сброс не дал результатов, в таком случае нужно вновь обесточить сервер.

После проделанных действий сервер будет перезапущен.

6) Сброс модуля удаленного доступа. Для этого потребуется при запуске сервера, в момент инициализации данного модуля нажать необходимую клавишу, отвечающую за переход в BIOS контроллера удаленного доступа. После инициализации, будет осуществлен переход в меню модуля, где будет предложен сброс.

После проделанных манипуляций по сбросу настроек оборудования, даем серверу загрузиться для дальнейшей диагностики.

7) Запуск необходимой утилиты, в зависимости от типа сервера, с Flash накопителя для его диагностики.

При всех манипуляциях, которые мы производили с сервером до данного момента, нами были физически продиагностированы все компоненты сервера и их работа в совокупности друг с другом. Но, до запуска диагностических утилит, мы не уверены в полной работоспособности сервера так как, нам нужно протестировать все компоненты программным способом чтобы сделать конечное заключение, а также, нам нужно зафиксировать работоспособность в документе с подтверждением работоспособности.

Для проведения тестов используется множество загрузочных Flash-накопителей с разным диагностическим ПО, а также установочными образами операционных систем и Live образов с набором программ.

В связи с увеличением поставляемого оборудования, возникла необходимость в оптимизации рабочих процессов. Так как, каждый инженер способен одновременно тестировать сразу несколько серверов, количество Flash-накопителей постоянно растет.

Используемые Flash-накопители показаны на рисунке 1.1 – 1.3



Рисунок 1.1 – Flash-накопители одного инженера



Рисунок 1.2 – Flash-накопители другого инженера



Рисунок 1.3 – Flash-накопители третьего инженера

В связи с большим количеством накопителей возникла необходимость использовать вместо накопителей другой ресурс.

## 1.2 Выбор метода разработки

Существует пара технологий, которые мы рассматривали в качестве альтернативы Flash-накопителям.

Первый вариант к рассмотрению, внешний бокс для HDD который имеет уникальную функцию – он может монтировать записанные в него образы дисков и дискет. То есть, кроме режима обычного внешнего HDD, он может работать в качестве DVD-ROM и флоппи-дисковод с загрузкой ISO образов напрямую.

Данный вариант имеет ряд плюсов, такие как:

- Готовое решение;
- Скорость передачи данных;
- Установка любого HDD.

Но также имеются и минусы данного решения, такие как:

- Один диск – один сервер;
- Непосредственное подключение к рабочей станции.

Вторым вариантом к рассмотрению стал сервер PXE. Так как выбор был сделан в пользу него, рассмотрим плюсы и минусы, а дальше перейдем к разбору данной технологии.

- Плюсы данного решения следующие:
- Можно использовать удаленно;
- Использование одновременно на нескольких рабочих станциях;
- Одновременное использование сразу несколькими инженерами;
- Гибкость в настройке;

Минусы данного решения;

- Не высокая скорость загрузки ПО;
- Разработка и настройка.

В связи с тем, что, имея один сервер PXE можно обеспечить загрузку одновременно сразу нескольких машин, а также работу могут производить сразу несколько инженеров, выбор был сделан в пользу PXE.

Загрузочный Flash-накопитель с набором нужных программ — замечательный инструмент системного администратора. Но лучше, чем загрузочный Flash-накопитель, это загрузочный сервер.

Представьте, вы выбрали в BIOS загрузку по сети и можете установить операционную систему, вылечить компьютер от вирусов, реанимировать диски, протестировать ОЗУ, с PXE Boot сервера, ведь это куда удобнее, нежели бегать с флешкой от машины к машине.

А в случае большого компьютерного парка, такой инструмент и вовсе незаменим.

PXE (Preboot eXecution Environment) – среда для загрузки компьютера с помощью сетевой карты без использования локальных носителей данных (жёсткого диска, USB-накопителя и т.п.). Для организации загрузки системы в PXE, используются протоколы IP, UDP, DHCP и TFTP.

PXE-код, обычно находящийся в ПЗУ сетевой карты, получает из сети по протоколу TFTP, изначально получив адрес TFTP-сервера по DHCP, загружает исполняемый файл, после чего передаёт ему управление.

Спецификация PXE ограничивает размер загрузчика 32 килобайтами, поэтому, иногда используется двух стадийная загрузка, когда первый загрузчик получает и запускает вторичный, который уже получает и запускает образ операционной системы.

Одна из реализаций загружаемого по PXE файла – pxelinux из комплекта программ Syslinux. Pxelinux умеет показывать пользователю меню, похожее на меню syslinux, а также загружать по сети и передавать управление ядру Linux и программам, имеющим идентичный формат загрузки (например, Memtest86). Сама сетевая загрузка выглядит следующим образом. В BIOS материнской платы активируется функция OnBoard LAN Boot ROM (данная функция как раз и включает исполнение кода PXE прошивки) и в качестве первого загрузочного устройства выбирается сетевая карта.

Во время загрузки, BIOS передает управление загрузочной микропрограмме сетевой карты. Цель данной программы, получить сетевые настройки (IP адрес, маску подсети, шлюз), адрес сервера с которого будет производится дальнейшая загрузка, и имя файла который следует загрузить с сервера и выполнить. Запрос, ожидание и принятие всех вышеописанных настроек осуществляется протоколом DHCP.

Протокол DHCP.

DHCP (Dynamic Host Configuration Protocol – протокол динамической настройки узла) – сетевой протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для



автоматической конфигурации компьютер-клиент, на этапе конфигурации сетевого устройства, обращается к серверу DHCP, и получает от него нужные параметры. Системный администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшить количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

DHCP является расширением протокола BOOTP, использовавшегося ранее для обеспечения бездисковых рабочих станций IP-адресами при их загрузке. DHCP сохраняет обратную совместимость с BOOTP.

Протокол DHCP предоставляет три способа распределения IP-адресов:

**Ручное распределение.** При этом способе, сетевой администратор сопоставляет аппаратному адресу (для Ethernet сетей это MAC-адрес), каждого клиентского компьютера, определённый IP-адрес. Фактически, данный способ распределения адресов, отличается от ручной настройки каждого компьютера лишь тем, что сведения об адресах хранятся централизованно на сервере DHCP, и потому их проще изменять при необходимости.

**Автоматическое распределение.** При данном способе, каждому компьютеру на постоянное использование выделяется произвольный свободный IP-адрес из определённого администратором диапазона.

**Динамическое распределение.** Этот способ аналогичен автоматическому распределению за исключением того, что адрес выдаётся компьютеру не на постоянное пользование, а на определённый срок. Это называется арендой адреса. По истечении срока аренды IP-адрес вновь считается свободным, и клиент обязан запросить новый. Кроме того, клиент сам может отказаться от полученного адреса.

Протокол DHCP является клиент-серверным, то есть в его работе участвуют клиент DHCP и сервер DHCP. Передача данных производится при помощи протокола UDP. По умолчанию запросы от клиента делаются на 67 порт к серверу, сервер в свою очередь отвечает на порт 68 к клиенту, выдавая адрес IP и другую необходимую информацию, такую, как сетевую маску, маршрутизатор и серверы DNS.

Все сообщения протокола DHCP разбиваются на поля, каждое из которых содержит определённую информацию. Все поля, кроме последнего имеют фиксированную длину. Описание полей указано в таблице 1.1.

Таблица 1.1 – поля протокола DHCP

Поле	Описание	Длина (в байтах)
op	Тип сообщения. Например может принимать значения: BOOTREQUEST (0x01, запрос от клиента к серверу) и BOOTREPLY (0x02, ответ от сервера к клиенту).	1
htype	Тип аппаратного адреса. Допустимые значения этого поля определены в RFC 1700 «Assigned Numbers». Например, для MAC-адреса Ethernet это поле принимает значение 0x01.	1
hlen	Длина аппаратного адреса в байтах. Для MAC-адреса Ethernet – 0x06.	1
hops	Количество промежуточных маршрутизаторов (так называемых агентов ретрансляции DHCP), через которые прошло сообщение. Клиент устанавливает это поле в 0x00.	1
xid	Уникальный идентификатор транзакции в 4 байта, генерируемый клиентом в начале процесса получения адреса.	4
secs	Время в секундах с момента начала процесса получения адреса. Может не использоваться (в этом случае оно устанавливается в 0x0000).	2
flags	Поле для флагов – специальных параметров протокола DHCP.	2
ciaddr	IP-адрес клиента. Заполняется только в том случае, если клиент уже имеет собственный IP-адрес и способен отвечать на запросы ARP (это возможно, если клиент выполняет процедуру обновления адреса по истечении срока аренды).	4



yiaddr	Новый IP-адрес клиента, предложенный сервером.	4
siaddr	IP-адрес сервера. Возвращается в предложении DHCP (см. ниже).	4
giaddr	IP-адрес агента ретрансляции, если таковой участвовал в процессе доставки сообщения DHCP до сервера.	4
chaddr	Аппаратный адрес (обычно MAC-адрес) клиента.	16
sname	Необязательное имя сервера в виде нуль-терминированной строки.	64
file	Необязательное имя файла на сервере, используемое бездисковыми рабочими станциями при удалённой загрузке. Как и sname, представлено в виде нуль-терминированной строки.	128
options	Поле опций DHCP. Здесь указываются различные дополнительные параметры конфигурации. В начале этого поля указываются четыре особых байта со значениями 99, 130, 83, 99 («волшебные числа»), позволяющие серверу определить наличие этого поля. Поле имеет переменную длину, однако DHCP-клиент должен быть готов принять DHCP-сообщение длиной в 576 байт (в этом сообщении поле options имеет длину 340 байт).	переменная

Рассмотрим пример процесса получения IP-адреса клиентом от сервера DHCP. Предположим, клиент ещё не имеет собственного IP-адреса, но ему известен его предыдущий адрес – 192.168.1.100. Процесс состоит из четырёх этапов. Эти этапы часто сокращаются как DORA (Discovery, Offer, Request, и Acknowledgement).

Обнаружение DHCP. В начале клиент выполняет широковещательный запрос по всей физической сети с целью обнаружить доступные DHCP-серверы. Он отправляет сообщение типа DHCPDISCOVER, при этом в качестве IP-адреса источника указывается 0.0.0.0 (если компьютер ещё не имеет собственного IP-

адреса), а в качестве адреса назначения – широковещательный адрес 255.255.255.255.

Клиент заполняет несколько полей сообщения начальными значениями:

- В поле `xid` помещается уникальный идентификатор транзакции, который позволяет отличать данный процесс получения IP-адреса от других, протекающих в то же время;
- В поле `chaddr` помещается аппаратный адрес (MAC-адрес) клиента;
- В поле опций указывается последний известный клиенту IP-адрес. В данном примере это 192.168.1.100. Это необязательно и может быть проигнорировано сервером.

Сообщение DHCPDISCOVER может быть распространено за пределы локальной физической сети при помощи специально настроенных агентов ретрансляции DHCP, перенаправляющих поступающие от клиентов сообщения DHCP серверам в других подсетях.

Не всегда процесс получения IP адреса начинается с DHCPDISCOVER. В случае, если клиент ранее уже получал IP адрес и срок его аренды ещё не прошёл – клиент может пропустить стадию DHCPDISCOVER начав с запроса DHCPREQUEST отправляемого с идентификатором сервера, который выдал адрес в прошлый раз. В случае же отсутствия ответа от DHCP сервера, выдавшего настройки в прошлый раз, клиент отправляет DHCPDISCOVER. Тем самым, клиент начинает процесс получения с начала, обращаясь уже ко всем DHCP серверам в сегменте сети.

Предложение DHCP. Получив сообщение от клиента, сервер определяет требуемую конфигурацию клиента в соответствии с указанными сетевым администратором настройками. В данном случае DHCP-сервер согласен с запрошенным клиентом адресом 192.168.1.100. Сервер отправляет ему ответ (DHCPOFFER), в котором предлагает конфигурацию. Предлагаемый клиенту IP-адрес указывается в поле `yaddr`. Прочие параметры (такие, как адреса маршрутизаторов и DNS-серверов) указываются в виде опций в соответствующем поле.

Это сообщение DHCP-сервер отправляет хосту, пославшему DHCPDISCOVER, на его MAC, при определенных обстоятельствах сообщение может распространяться как широковещательная рассылка. Клиент может получить несколько различных предложений DHCP от разных серверов, из них он должен выбрать то, которое его «устраивает».

Запрос DHCP. Выбрав одну из конфигураций, предложенных DHCP-серверами, клиент отправляет запрос DHCP (DHCPREQUEST). Он рассылается широковещательно, при этом к опциям, указанным клиентом в сообщении DHCPDISCOVER, добавляется специальная опция – идентификатор сервера, указывающая адрес DHCP-сервера, выбранного клиентом.

Подтверждение DHCP. Наконец, сервер подтверждает запрос и направляет это подтверждение (DHCPACK) клиенту. После этого клиент должен настроить свой сетевой интерфейс, используя предоставленные опции.

После того как микропрограмма сетевой карты запустилась, она посылает широковещательный запрос, с целью найти DHCP сервер, который предоставит все вышеописанные настройки. Это означает, что в сети должен присутствовать DHCP сервер. При наличии DHCP сервера, и получении от него ответа, микропрограмма сетевой карты примет переданные ей настройки. А именно установит сетевые параметры (IP адрес, маску подсети, шлюз), после чего попытается подключиться к серверу загрузки по протоколу TFTP и загрузить с него указанный исполняемый файл. То есть, кроме DHCP сервера в сети еще должен присутствовать TFTP сервер.

При наличии TFTP сервера, и присутствии на нем указанного файла, микропрограмма сетевой карты загрузит и попытается его выполнить. Если данный файл действительно исполняемый, то дальнейшая загрузка будет продолжаться согласно инструкциям, прописанным в данном файле. Обычно данным файлом является загрузчик (bootmgr, grub4dos, syslinux, pxelinux, efigrub) которому будет передано дальнейшее управление.

### Протокол TFTP.

TFTP (англ. Trivial File Transfer Protocol – простой протокол передачи

файлов) используется главным образом для первоначальной загрузки бездисковых рабочих станций. TFTP, в отличие от FTP, не содержит возможностей аутентификации и основан на транспортном протоколе UDP.

Основное назначение TFTP – обеспечение простоты реализации клиента. В связи с этим, он используется для загрузки бездисковых рабочих станций, загрузки обновлений и конфигураций в «умные» сетевые устройства, записи статистики с мини-АТС (CDR) и аппаратных маршрутизаторов/межсетевых экранов.

Поскольку протокол не поддерживает аутентификации, единственный метод идентификации клиента – это его сетевой адрес, который может быть подделан. Обычно, в Unix-системах, tftpd доступен только каталог /tftpboot. Однако в старых TFTP-серверах было возможным получить файл паролей командой «RRQ ../etc/passwd».

Демон tftpd (одна из реализаций tftp-сервера) отказывается обрабатывать файлы, содержащие в своём имени комбинацию «/./» или начинающуюся с «./». Запись разрешается только в файлы, которые уже существуют, любого размера, и доступны для публичной записи права доступа: -rw-rw-rw-.

Дополнительная защита от доступа к произвольным файлам осуществляется с помощью смены корневого каталога на каталог tftpd.

Сначала в TFTP-пакете идет поле размером в 2 байта, определяющее тип пакета:

- Read Request (RRQ, #1) – запрос на чтение файла;
- Write Request (WRQ, #2) – запрос на запись файла;
- Data (DATA, #3) – данные, передаваемые через TFTP;
- Acknowledgment (ACK, #4) – подтверждение пакета;
- Error (ERR, #5) – ошибка.
- Option Acknowledgment (OACK, #6) – подтверждение опций.

Для начала передачи данных, клиент должен послать серверу WRQ или RRQ-пакет. У обоих пакетов формат одинаковый:

- 0x01/0x02 (тип пакета) – Имя файла;
- 0x00 (конец строки) – Режим передачи;

– 0x00 (конец строки) – Опции.

После получения RRQ-пакета сервером, он сразу начинает передачу данных. В случае с WRQ-запросом – сервер должен прислать ACK-пакет с номером пакета 0.

После получения запроса RRQ, сервер сразу посылает, в качестве подтверждения пакет с данными и с ID пакета, равным единице. В WRQ в качестве подтверждения используется ACK с ID, равным нулю. Всего по TFTP можно передать 32Мб, однако, из-за использования знакового int вместо без знакового, размер подтверждения ограничен 16 мегабайтами. Однако, если клиент и сервер поддерживают расширения протокола RFC 2347 и RFC 2348, то максимальный размер передаваемого файла увеличивается до 4Gb.

Web-сервер lighttpd.

Web-сервером называют как программное обеспечение, выполняющее его функции, так и сам компьютер или сервер, на котором это программное обеспечение работает.

Web-сервер, это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, но в нашем случае запросы будут поступать от сервиса ipxe, и выдавать им HTTP-ответы, после этого будет происходить сразу передача файлов. Работает это следующим образом:

Клиент передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы, это HTML-страницы, изображения, файлы, медиа-потoki или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

HTTP это протокол прикладного уровня. Обмен сообщениями идёт по схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные URL.

В роли нашего web-сервера будет выступать lighttpd.

lighttpd один из веб-серверов, разрабатываемый и оптимизированный с расчётом на работу в требовательных к скорости средах, не теряя при этом безопасности и гибкости, а также совместимости со стандартами.

Выбор был сделан исходя из следующих преимуществ:

Низкий объем памяти (по сравнению с другими веб-серверами), небольшая загрузка процессора и оптимизация скорости делают `lighttpd` подходящим для серверов, которые испытывают проблемы с загрузкой, или для обслуживания статических носителей отдельно от динамического контента.

`lighttpd` использует так называемую асинхронную обработку сетевых соединений. Благодаря этому загруженность сервера, в отличие от `Apache`, при доступе к файлам на диске не зависит от количества текущих соединений.

В `lighttpd` возможно использование особых системных вызовов для повышения производительности при передаче файлов. При этом задействуются не стандартные системные интерфейсы, а специфичные для платформы вызовы ядра операционной системы, и, таким образом, смена контекста CPU сводится к минимуму, что необходимо при использовании нами одноплатного компьютера.

Web-сервер необходим нам для передачи массивных файлов. Так как скорость через протокол `FTTPD` низкая, а с помощью web-сервера мы сможем загружать ISO образы наших систем или программ.

Взаимодействие протоколов для работы PXE.

Сама сетевая загрузка выглядит следующим образом. В BIOS материнской платы активируется функция `OnBoard LAN Boot ROM` (данная функция как раз и включает исполнение кода PXE прошивки) и в качестве первого загрузочного устройства выбирается сетевая карта.

Во время загрузки, BIOS передает управление загрузочной микропрограмме сетевой карты. Цель данной программы, получить сетевые настройки (IP адрес, маску подсети, шлюз), адрес сервера с которого будет производиться дальнейшая загрузка, и имя файла который следует загрузить с сервера и выполнить. Запрос, ожидание и принятие всех вышеописанных настроек осуществляется протоколом DHCP.

После того как микропрограмма сетевой карты запустилась, она посылает широковещательный запрос, с целью найти DHCP сервер, который предоставит все вышеописанные настройки. Это означает, что в сети должен присутствовать DHCP сервер. При наличии DHCP сервера, и получении от него ответа, микропрограмма

сетевой карты примет переданные ей настройки. А именно установит сетевые параметры (IP адрес, маску подсети, шлюз), после чего попытается подключиться к серверу загрузки по протоколу TFTP и загрузить с него указанный исполняемый файл. То есть, кроме DHCP сервера в сети еще должен присутствовать TFTP сервер.

При наличии TFTP сервера, и присутствии на нем указанного файла, микропрограмма сетевой карты загрузит и попытается его выполнить. Если данный файл действительно исполняемый, то дальнейшая загрузка будет продолжаться согласно инструкциям, прописанным в данном файле. Обычно данным файлом является загрузчик (bootmgr, grub4dos, syslinux, pxelinux, efigrub) которому будет передано дальнейшее управление.

### **1.3 Формализованное описание технического задания**

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы».

#### **1. Общие сведения.**

##### **1.1. Название организации-заказчика.**

ООО «НАГ»

##### **1.2. Название продукта разработки.**

«Сервер PXE с удаленным доступом»

##### **1.3. Назначение продукта.**

Данный сервер разработан для технического отдела компании ООО «НАГ», в связи с необходимостью сократить объем используемых накопителей, а также оптимизировать время работы инженеров и централизовать используемое программное обеспечение на одном сервере для удобного обслуживания, администрирования и внесения изменений в конфигурацию.

##### **1.4. Плановые сроки начала и окончания работ.**

В соответствии с планом выполнения ВКР.

#### **2. Характеристика области применения продукта.**

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

РХЕ-сервер предполагается использовать инженерами выше указанной компании, для проведения диагностических работ, входящих в проведение предпродажной подготовки серверного оборудования.

## 2.2. Характеристика персонала

2.3. Инженеры технического отдела выполняющие предпродажные подготовки серверного оборудования. Готовность к работе с данным сервером имеется.

## 3. Требования к продукту разработки.

### 3.1. Требования к продукту в целом.

Данный сервер должен обеспечить работу сразу нескольким инженерам одновременно, а также обеспечить достаточную скорость загрузки программного обеспечения для сокращения времени выполнения тестов.

### 3.2. Аппаратные требования.

Минимальные аппаратные характеристики должны быть выше или соответствовать следующим:

- CPU – двухъядерный процессор на архитектуре ARM или x86 с тактовой частотой 1.5GHz;
- ROM – 512 MB;
- Внутренняя энергонезависимая память EMMC – 4GB;
- Внешний Flash накопитель MicroCD – 16GB;
- Питание 5В 2А.

3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

Программное обеспечение базируется на системе Linux.

3.4. Указание программного обеспечения, используемого для реализации.

Для монтирования системы на носитель, так как это не стандартный компьютер, а одноплатный на процессоре ARM, нам потребуется программа BalenaEtcher.



3.5. Для сетевых систем – особенности реализации серверной и клиентской частей.

Для реализации серверной части нам потребуется установка служб TFTP, DHCP и Lighttpd. Для клиентской части нам ничего не нужно устанавливать, так как серверная часть все сделает автоматически.

3.6. Форматы входных и выходных данных

Не предусмотрено.

3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Не предусмотрено.

3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Не предусмотрено.

3.9. Меры защиты информации.

Не предусмотрено.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

Пользовательский интерфейс должен включать в себя все необходимые пункты меню для установки или запуска необходимого ПО.

4.2. Размещение информации на экране, дизайн экрана.

Информация на экране выглядит следующим образом, как показано на рисунке 1.6

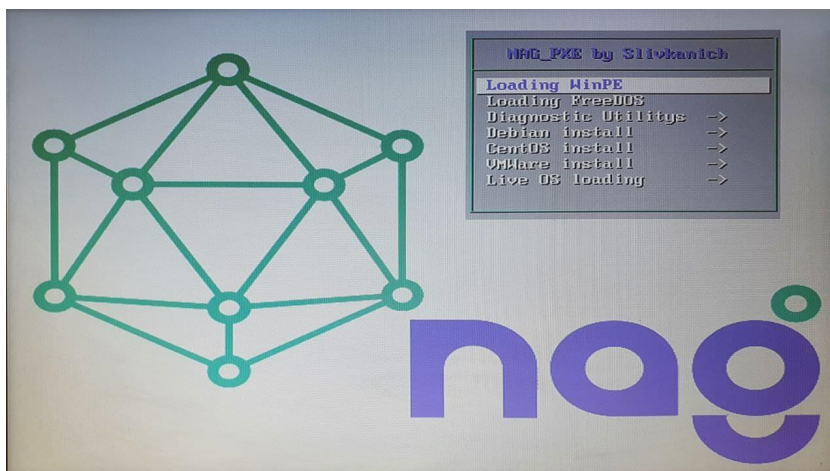


Рисунок 1.6 – Дизайн загрузчика

## Глава 2. Практическая часть

### 2.1 Модельные представления объекта разработки

Для небольшого сервера DHCP с файловым обменом по средствам TFTP для реализации загрузчика PXE, подойдет даже одноплатный компьютер Orange PI PC Plus с характеристиками:

- CPU – двух ядерный процессор на архитектуре ARM с тактовой частотой 1.8GHz;
- ROM – 1024 MB;
- Внутренняя энергонезависимая память EMMC – 8GB;
- Внешний Flash накопитель MicroSD – 16GB.
- Питание 5В 2А.

Основанием для выбора стало то, что сервисы для которых реализуется данный проект в виде мобильной станции, обеспечивают гораздо больше удобства для системного администрирования, чем статическая станция в серверной.

Приступим к установке системы на одноплатный компьютер, для этого не понадобится первоначальная настройка самого компьютера, нужно развернуть систему непосредственно на microSD ноутбука или компьютера под управлением OS Windows.

Для моделирования аналогичных настроек необходимых сервисов, была выбрана операционная система Linux система для одноплатных компьютеров Armbian, разработанная на ядре Debian под архитектуру мобильных процессоров ARM.

Скачав операционную систему с официального сайта, предварительно подключив накопитель к компьютеру, приступим к развертыванию образа системы на Flash microSD при помощи специальной утилиты BalenaEtcher. Ход установки операционной системы показан на рисунках 2.1 – 2.5.

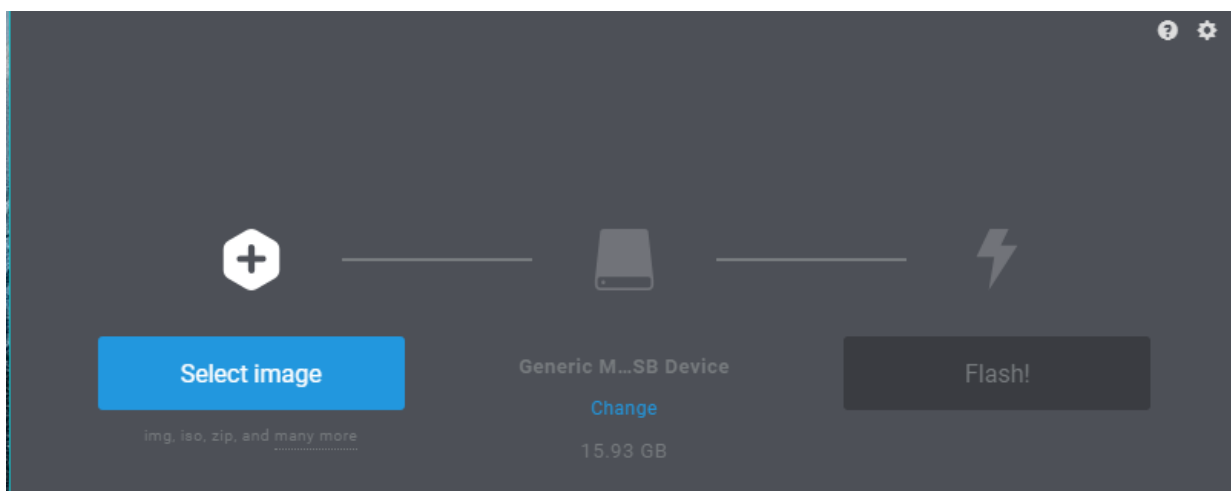


Рисунок 2.1 – Пункт выбора образа системы

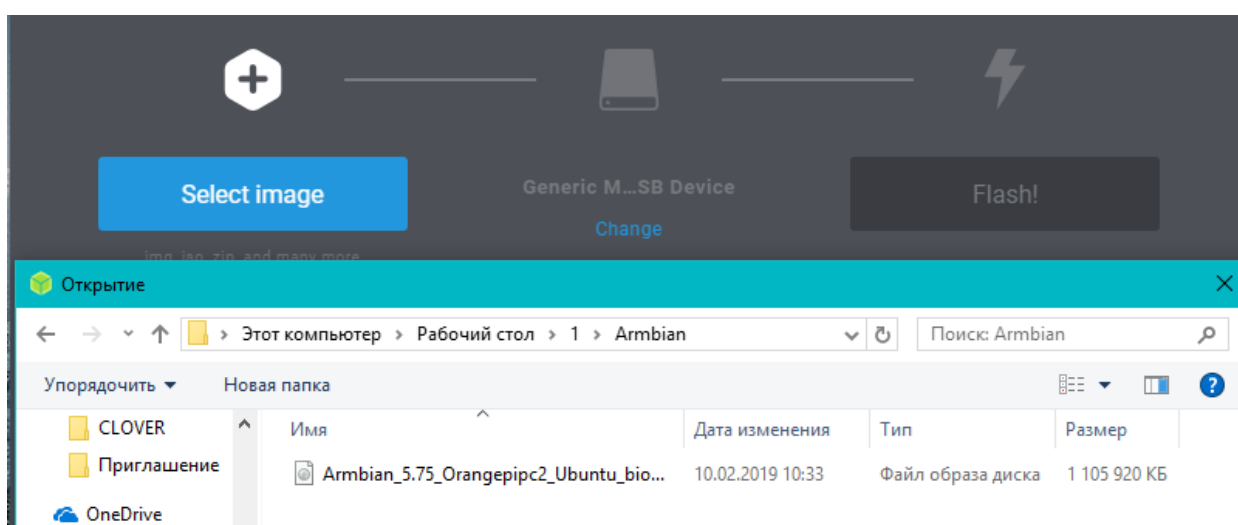


Рисунок 2.2 – Выбор операционной системы для монтирования образа

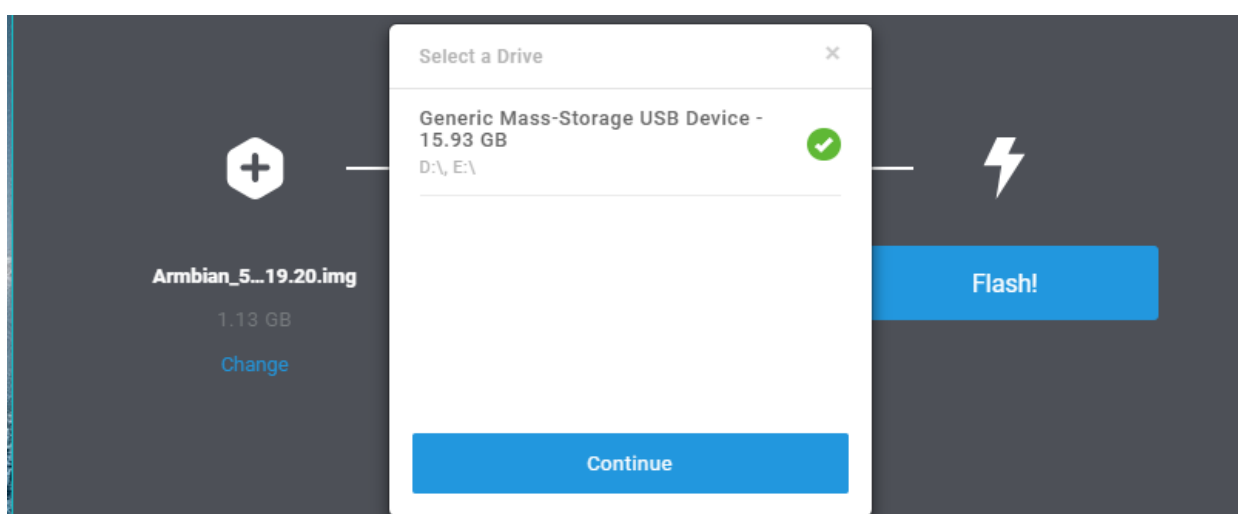


Рисунок 2.3 – Выбор накопителя для создания загрузчика системы

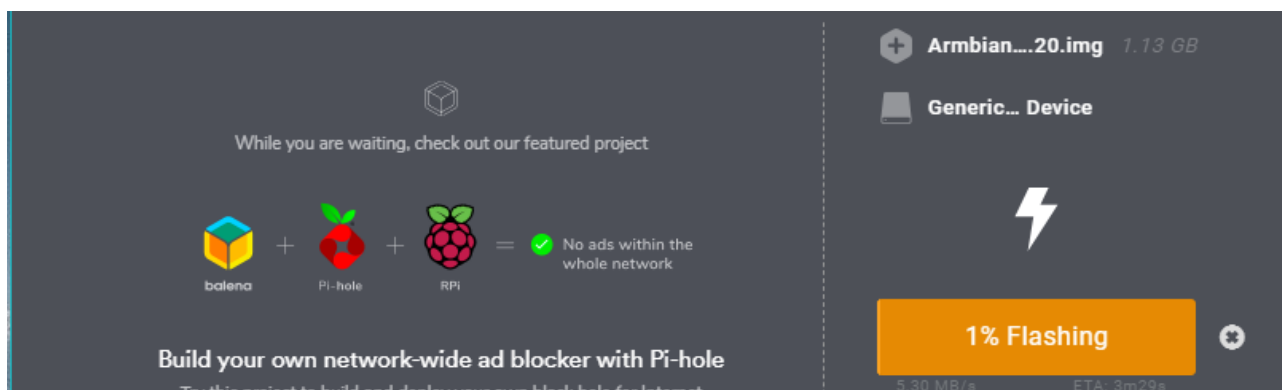


Рисунок 2.4 – Ход развертывания системы на накопитель

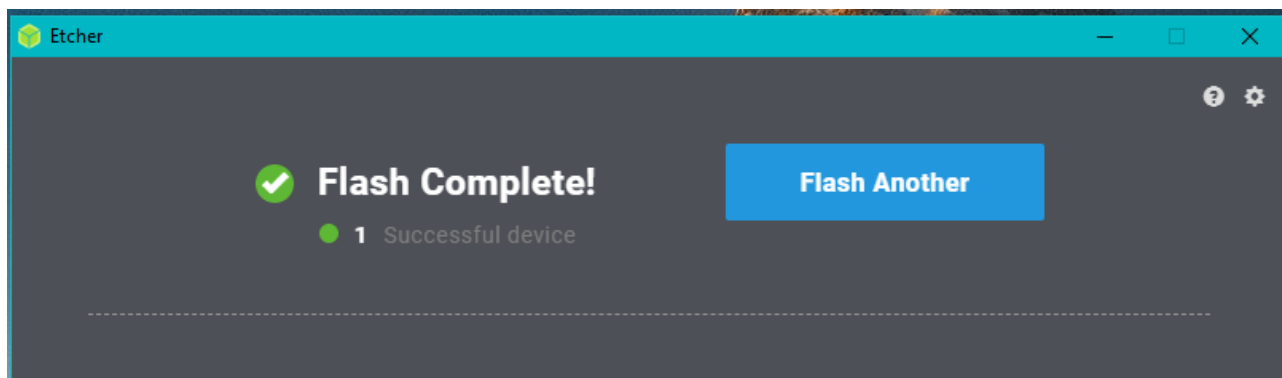


Рисунок 2.5 – Успешное выполнение по развертыванию системы на накопитель

После переноса системы на Flash накопитель, устанавливаем накопитель в одноплатный компьютер, подключаем монитор, клавиатуру и питание.

После включения и полной загрузки операционной системы переходим к установке и настройке необходимых служб.

Перед началом установки сервисов, необходимо подготовить нашу машину для их установки, что бы в дальнейшем не возникло ошибок с позиционированием файлов и их доступности для сервисов.

Первым делом создаем директорию, к которой необходимо будет предоставить доступ сервисам TFTPd, DHCP, и Lighttpd.

Создаем директорию для которой будем предоставлять доступ сервисам:

```
mkdir -p /PXEBOOT/HTTPLoad
```

Далее производим обновление системы по средствам команд:

```
apt-get update
```

```
apt-get upgrade
```

Для корректного запуска сервиса DHCP, необходимо настроить наш сетевой интерфейс следующим образом.

Первым шагом, открываем файл конфигурации сетевых интерфейсов с используя редактор текста “NANO”:

```
nano /etc/network/interfaces
```

Открывшийся конфигурационный файл приводим к следующему содержимому:

Задаем интерфейсу eth0 запуск автоматически при старте системы.

```
auto eth0
```

Задаем метод адресации статический.

```
iface eth0 inet static
```

Задаем IP адрес.

```
address 10.10.10.14
```

Задаем адрес сети.

```
gateway 10.10.10.1
```

Задаем маску сети.

```
netmask 255.255.255.0
```

После проведенных манипуляций переходим к установке первого сервиса tftpd-hpa.

Установка и настройка сервиса tftpd-hpa.

Установку сервиса tftpd-hpa производим следующей командой:

```
apt-get install tftpd-hpa
```

После установки сервиса необходимо его сконфигурировать, для этого переходим в конфигурационный файл данного сервиса:

```
nano /etc/default/tftpd-hpa
```

Открывшийся конфигурационный файл приводим к следующему содержимому:

Задаем имя сервера.

```
TFTP_USERNAME="tftp"
```

Прописываем директорию в которой будут расположены все файлы для работы PXE сервера.

```
TFTP_DIRECTORY="/PXEBOOT"
```

Прописываем локальный адрес сети с портом доступа по TFTP.

```
TFTP_ADDRESS="0.0.0.0:69"
```

Оставляем опции по умолчанию.

```
TFTP_OPTIONS="--ipv4 --secure --create"
```

При помощи данного сервиса мы будем передавать на клиентские машины необходимые файлы загрузки, находящиеся в директории – `"/PXEBOOT"`.

Для применения внесенных нами изменений в сервис, производим его перезагрузку командой:

```
service tftpd-hpa restart
```

После перезагрузки сервиса, переходим к установке и настройке следующего.

Установка и настройка сервиса DHCP.

Установку сервиса `isc-dhcp-server` производим следующей командой:

```
apt-get install isc-dhcp-server
```

После установки сервиса необходимо его сконфигурировать, для этого переходим в конфигурационный файл данного сервиса:

```
nano /etc/dhcp/dhcpd.conf
```

Открывшийся конфигурационный файл приводим к следующему содержимому:

Тип авторизации при подключении клиента к серверу;

```
authoritative;
```

Указание сети, в которой будет работать DHCP;

```
subnet 192.168.1.0 netmask 255.255.255.0
```

Создание пула адресов для раздачи клиентам;

```
range 192.168.1.70 192.168.1.80;
```

Указание имени сервера;

*option domain-name "localdomain";*

Указание адреса сервера;

*option domain-name-servers 192.168.1.10;*

Указание широковещательного адреса сети;

*option broadcast-address 192.168.1.255;*

Указание адреса маршрутизатора;

*option routers 192.168.1.1;*

Указание маски подсети;

*option subnet-mask 255.255.255.0;*

Указание пути расположения файлов операционной системы и файла загрузки PXE;

*option root path "/opt/netboot";*

Указание имени сервера выступающего в роли сервера DHCP;

*server-name "192.168.1.10";*

*next-server 192.168.1.10;*

Указание имени файла, который будет выступать в роли загрузочного.

*filename "pxelinux.0";*

Где мы описываем параметры нашего сетевого интерфейса для работы с ним данной службы, а также указываем параметры для аренды IP адресов и первоначальный файл загрузки сервиса PXE.

После того, как данный файл был сконфигурирован, переходим к указанию данной службе, с каким сетевым интерфейсом ей предстоит работать.

Для этого переходим в конфигурационный файл следующей командой:

*nano /etc/default/isc-dhcp-server*

Далее приводим его к следующему содержимому:

Указываем файл конфигурации сетевых реквизитов.

DHCPDv4\_CONF=/etc/dhcp/dhcpd.conf

Указываем имя сетевого интерфейса.

INTERFACESv4="eth0"

При помощи данного сервиса мы будем предоставлять в аренду сетевые реквизиты по которым клиентские машины будут принимать нужные файлы для загрузки через PXE.

Для применения внесенных нами изменений в сервис, производим его перезагрузку командой:

```
service isc-dhcp-server restart
```

После перезагрузки сервиса, переходим к установке и настройке последнего, нужного там сервиса для полного функционирования сервера PXE.

Установка и настройка сервиса lighttpd.

Установку сервиса lighttpd производим следующей командой:

```
apt-get install lighttpd -y
```

Далее подключаем необходимые модули к сервису для его корректной работы с директориями:

```
lighty-enable-mod dir-listing
```

После установки сервиса необходимо его сконфигурировать, для этого переходим в конфигурационный файл данного сервиса:

```
nano /etc/lighttpd/lighttpd.conf
```

В открывшемся конфигурационном файле находим и изменяем следующую строчку:

```
server.document-root = "/PXEBOOT/HTTPLoad"
```

Где в параметр server.document-root, мы прописываем путь до директории в которой будут храниться массивные файлы для загрузки по HTTP протоколу.

Данный сервис необходим для быстрой загрузки файлов по протоколу HTTP, так как TFTP сервер не способен обеспечить нужную нам скорость передачи, но TFTP протокол необходим для начальной загрузки файлов загрузчиков как PXE-сервера, так и загрузочных файлов операционных систем, так как в первоначальной загрузке протокол HTTP не используется. При помощи данного сервиса мы будем передавать на клиентские машины необходимые файлы загрузки, находящиеся в



директории – "/PXEBOOT/HTTPLoad". Для применения внесенных нами изменений в сервис, производим его перезагрузку командой:

```
service lighttpd restart
```

После перезагрузки сервиса, переходим к заполнению необходимыми файлами созданные директории.

Создание системы файлов.

Создаем файлы конфигурации PXE сервера, а также заполняем директории диагностическими утилитами и установочными операционных систем, которые необходимы нам в работе.

После создания всех необходимых файлов и заполнения директорий диагностическими утилитами и установочными операционных систем, дерево каталогов выглядит так как показано на рисунке 2.8.

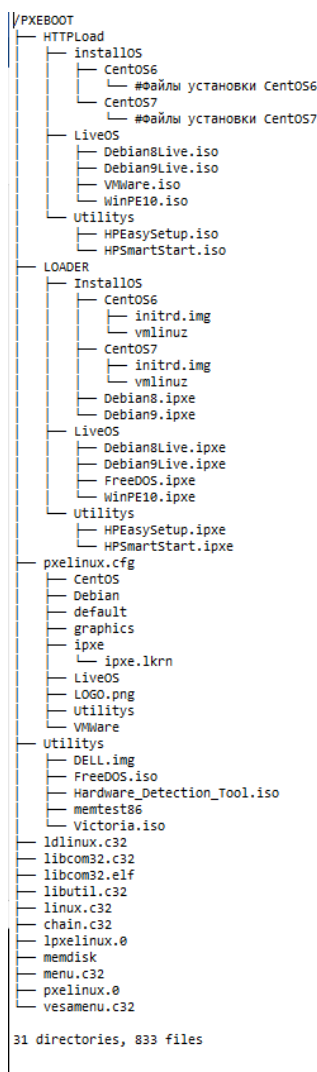


Рисунок 2.8 – Дерево каталогов

В корне директории /PXEBOOT хранятся файлы сервера PXE, а именно: `ldlinux.c32`, `libcom32.c32`, `libcom32.elf`, `libutil.c32`, `linux.c32`, `chain.c32`, `lpxelinux.0`, `memdisk`, `menu.c32`, `pxelinux.0`, `vesamenu.c32`.

Данные файлы представляют собой основные компоненты сервера PXE, а именно, это библиотеки, модули и прочие системные файлы сервиса.

Также, в корне каталога /PXEBOOT, расположена директория HTTPLoad, в которой хранятся массивные образы утилит и операционных систем, которые будут загружаться по протоколу HTTP.

Директория LOADER, в корне каталога PXEBOOT хранит в себе загрузчики тех массивных файлов, которые расположены в директории HTTPLoad.

Каталог Utilitys, хранит в себе не массивные утилиты, которые без ущерба времени загрузки способны быстро загрузиться при помощи протокола TFTP.

Таким образом, файлы из каталога LOADER загружаются на прямую при помощи протокола TFTP, после этого подгружают остальные файлы, находящиеся в каталоге HTTPLoad по протоколу HTTP, когда утилиты из каталога Utilitys загружаются полностью в память клиентской машины по средствам протокола TFTP.

Далее, переходим к каталогу pxelinux.cfg. Данный каталог хранит в себе все конфигурационные файлы PXE сервера, а также его графическую составляющую.

Разберем каждый файл данного каталога подробнее, так как он является основным каталогом пользовательского интерфейса, а также включает в себя конфигурацию загрузочных ссылок на диагностические утилиты или программное обеспечение.

Конфигурационный файл `default`.

Он предназначен для первоначального меню, где будут находиться ссылки на остальные подменю.

В первую очередь подключаем модуль загрузки графического меню.

*`default vesamenu.c32`*

Подключение файла с конфигурацией графического интерфейса.

*MENU INCLUDE pxelinux.cfg/graphics*

Прописываем название подменю.

*menu title NAG\_PXE by Slivkanich*

Прописываем блок для запуска WinPE, где указано отображаемое имя, подключен модуль для загрузки через ipxe и путь к конфигурационному файлу ipxe в котором прописаны необходимые параметры для загрузки WinPE.

*LABEL Loading WinPE*

*KERNEL pxelinux.cfg/ipxe/ipxe.lkrn*

*INITRD /LOADER/LiveOS/WinPE10.ipxe*

Аналогичный блок для загрузки, только для FreeDOS.

*LABEL Loading FreeDOS*

*KERNEL pxelinux.cfg/ipxe/ipxe.lkrn*

*INITRD /LOADER/LiveOS/FreeDOS.ipxe*

Прописываем блоки для загрузки подменю в соответствии с их содержанием, где первой строчкой прописываем наименование подменю, вторым пунктом подключаем графическую составляющую для подменю и следующим пунктом прописываем путь к файлу с конфигурацией данного подменю.

Подменю для диагностических утилит.

*LABEL Diagnostic Utilitys ->*

*KERNEL vesamenu.c32*

*APPEND pxelinux.cfg/Utilitys*

Подменю для установочных файлов Debian.

*LABEL Debian install ->*

*KERNEL vesamenu.c32*

*APPEND pxelinux.cfg/Debian*

Подменю для установочных файлов CentOS.

*LABEL CentOS install ->*

*KERNEL vesamenu.c32*

*APPEND pxelinux.cfg/CentOS*

Подменю для установочных файлов VMWare.

*LABEL VMWare install ->*

*KERNEL vesamenu.c32*

*APPEND pxelinux.cfg/VMWare*

Подменю для загрузочных файлов LiveOS.

*LABEL Live OS loading ->*

*KERNEL vesamenu.c32*

*APPEND pxelinux.cfg/LiveOS*

В следствии данных конфигураций загрузочный экран будет выглядеть как показано на рисунке 2.9.

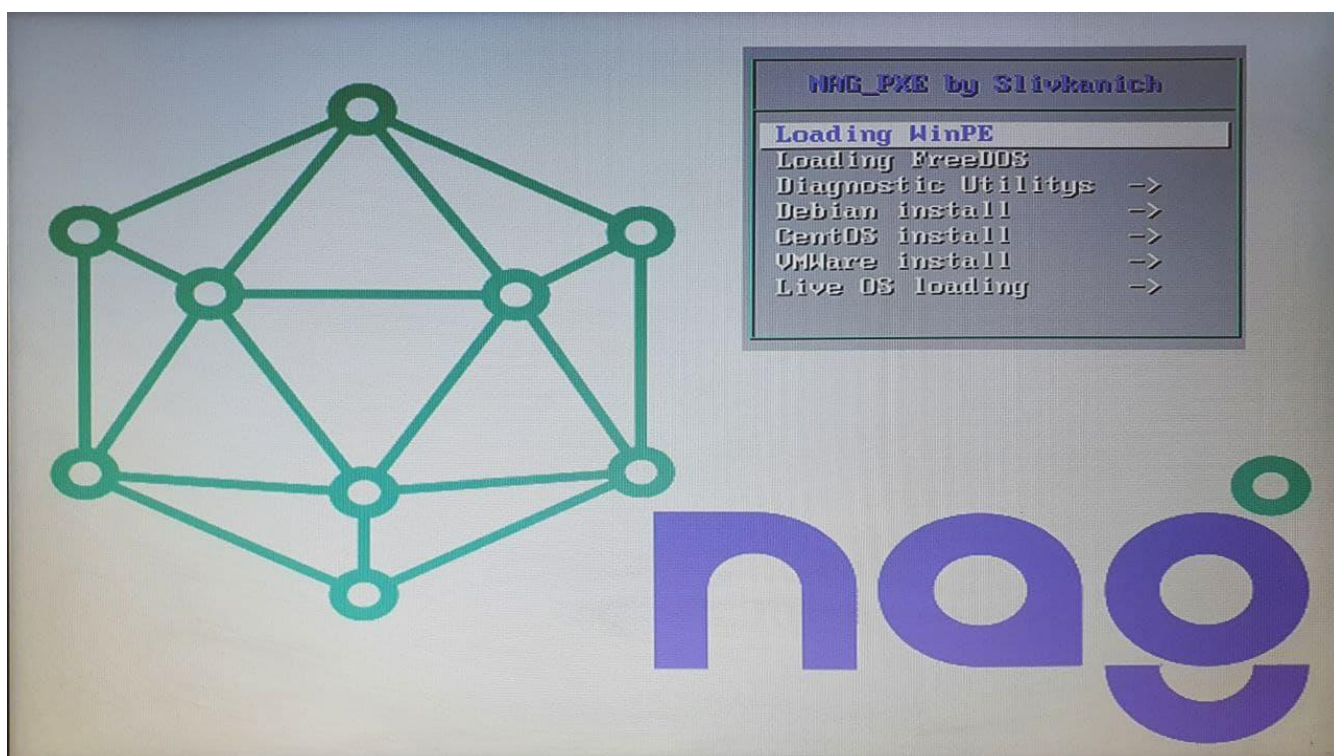


Рисунок 2.9 – Загрузочный экран default

Конфигурационный файл graphics.

Данный файл создается для централизованной настройки графических параметров как основного меню, так и второстепенных подменю.

Настройки задаем следующие:

Подключаем основной фон и прописываем путь до изображения.

*MENU BACKGROUND pxelinux.cfg/LOGO.png*

В данном блоке указываем параметры положения основного окна, а также его размеры.

*MENU WIDTH 50*

*MENU VSHIFT 1*

*MENU HSHIFT 33*

*MENU MARGIN 10*

*MENU ROWS 8*

В этом блоке указываем стиль и цвет каждого элемента меню.

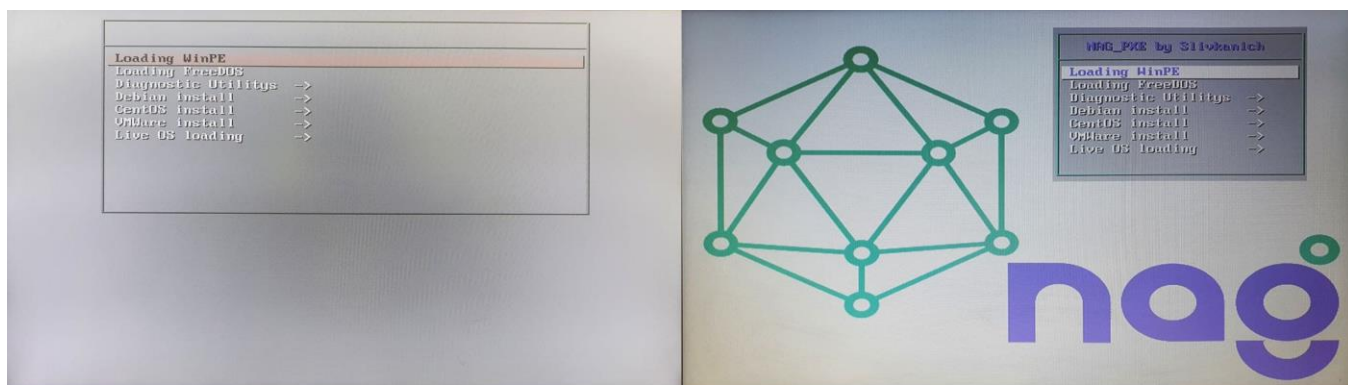
*MENU COLOR border 0 #66cdaa #60000000 std*

*MENU COLOR title 0 #9932CC #60000000 std*

*MENU COLOR sel 0 #9932CC #20ffffff all*

*MENU COLOR unsel 0 #50ffffff #60000000 std*

На рисунке 2.10 приведен пример меню с применёнными стилями оформления и без них.



Без графических настроек

С графическими настройками

Рисунок 2.10 – Сравнение

Все элементы оформления выпилены в соответствии с цветами компании и приведены к единому стилю оформления.

После настройки основного меню и применения стилей оформления переходим к настройке файлов подменю. Конфигурационные файлы подменю однотипны между собой за исключением названий и ссылок на загрузочные файлы.

Конфигурационный файл Debian.

В данном подменю содержатся ссылки на установочные образы операционных систем Debian 8 и Debian 9.

Подключаем графический модуль и прописываем название подменю.

```
default vesamenu.c32
```

```
MENU INCLUDE /pxelinux.cfg/graphics
```

```
MENU TITLE Debian
```

Прописываем название Debian 8 install, подключаем ядро загрузки для ipxe и указываем ссылку на файл загрузки ipxe для Debian 8.

```
LABEL Debian 8 install
```

```
kernel pxelinux.cfg/ipxe/ipxe.lkrn
```

```
initrd /LOADER/InstallOS/Debian8.ipxe
```

Аналогичный блок как для Debian 8, только указываем соответствующую ссылку на Debian 9.

```
LABEL Debian 9 install
```

```
kernel pxelinux.cfg/ipxe/ipxe.lkrn
```

```
initrd /LOADER/InstallOS/Debian9.ipxe
```

Блок возврата в основное меню.

```
LABEL <- Main Menu
```

```
KERNEL vesamenu.c32
```

```
APPEND pxelinux.cfg/default
```

В следствии данных конфигураций загрузочный экран будет выглядеть как показано на рисунке 2.11.



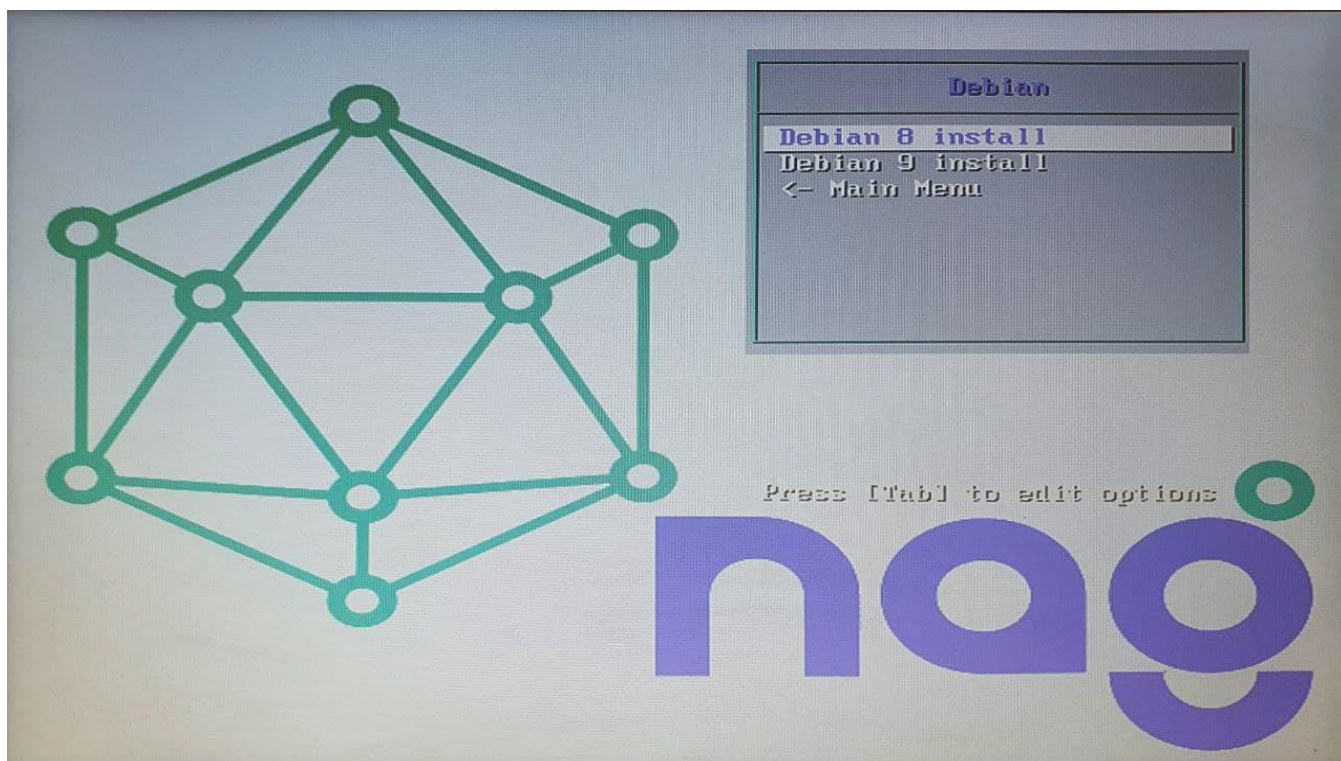


Рисунок 2.11 – Загрузочный экран Debian

Конфигурационный файл CentOS.

В данном подменю содержатся ссылки на установочные образы операционных систем CentOS 6 и CentOS 7.

Аналогичным образом подключаем графическую составляющую, как в предыдущих меню.

```
default vesamenu.c32
```

```
MENU INCLUDE /pxelinux.cfg/graphics
```

```
MENU TITLE CentOS
```

Так как, для операционных систем CentOS характерен другой метод установки, все файлы операционной системы находятся не в ISO образе, для их загрузки требуется загрузить Linux ядро, а после этого указать параметры установки и ссылки как на загрузчик системы, так и на расположение основных ее файлов.

```
LABEL CentOS 6
```

```
Kernel /LOADER/CentOS6/vmlinuz
```

```
Append    initrd=/LOADER/installOS/CentOS6/initrd.img
method=http://10.10.10.14/installOS/CentOS6/    devfs=nomount    ksdevice=link
ramdisk_size=100000
```

Аналогичным образом прописываем параметры для CentOS 7.

```
LABEL CentOS 7
```

```
Kernel /LOADER/CentOS7/vmlinuz
```

```
Append                                initrd=/LOADER/installOS/CentOS7/initrd.img
method=http://10.10.10.14/installOS/CentOS7/    devfs=nomount    ksdevice=link
ramdisk_size=100000
```

Блок возврата в основное меню аналогичен.

```
LABEL <- Main Menu
```

```
KERNEL vesamenu.c32
```

```
APPEND pxelinux.cfg/default
```

В следствии данных конфигураций загрузочный экран будет выглядеть как показано на рисунке 2.12.

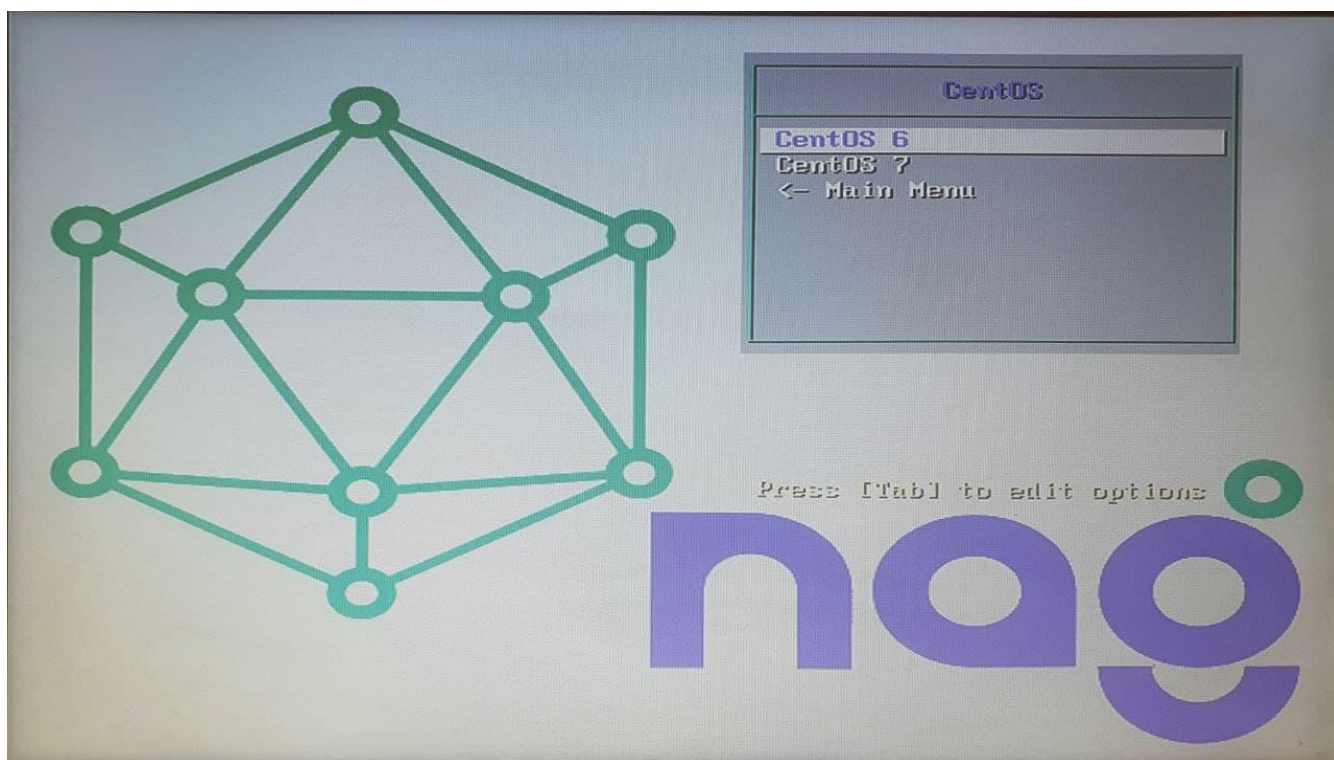


Рисунок 2.12 – Загрузочный экран CentOS

Конфигурационный файл LiveOS.



Конфигурационный файл LiveOS аналогичен по структуре с конфигурационным файлом Debian, за исключением ссылок на загрузочные файлы Live образов операционных систем, а не их установочные файлы.

```
default vesamenu.c32
MENU INCLUDE /pxelinux.cfg/graphics
MENU TITLE LiveOS
LABEL Debian 8 Live
kernel pxelinux.cfg/ipxe/ipxe.lkrn
initrd /LOADER/LiveOS/Debian8Live.ipxe
LABEL Debian 9 Live
kernel pxelinux.cfg/ipxe/ipxe.lkrn
initrd /LOADER/LiveOS/Debian9Live.ipxe
LABEL <- Main Menu
KERNEL vesamenu.c32
APPEND pxelinux.cfg/default
```

Исходя из этого, подменю выглядит так, как показано на рисунке 2.13.

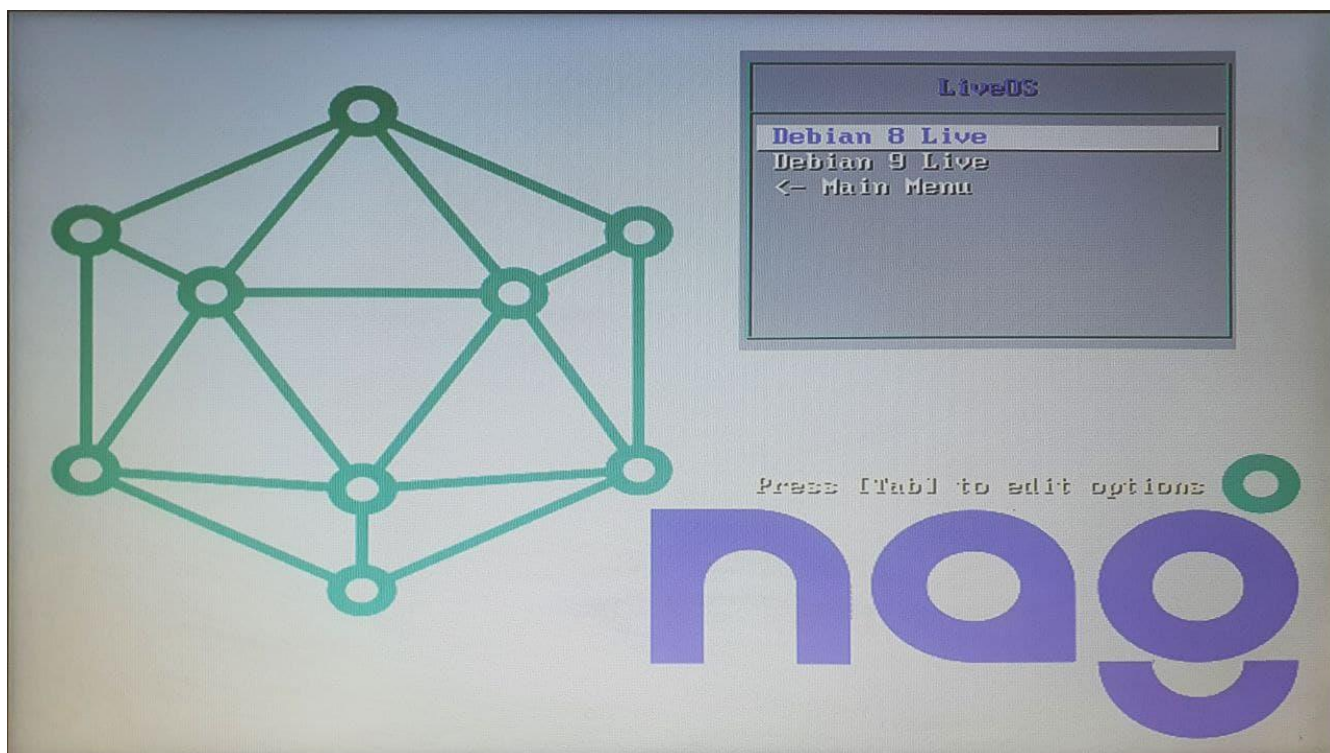


Рисунок 2.13 – Загрузочный экран LiveOS

Конфигурационный файл *Utility*s.

В данном конфигурационном файле собраны диагностические утилиты, которые постоянно используются для диагностики тех или иных серверных систем. Исходя из размера утилиты, используется разный метод загрузки.

Первым делом, по аналогии с предыдущими файлами, подключаем графическую составляющую.

```
default vesamenu.c32  
MENU INCLUDE /pxelinux.cfg/graphics  
MENU TITLE Utilitys
```

В данном блоке находится утилита для диагностики серверов HP до восьмого поколения. Для данной утилиты используется метод загрузки при помощи *ipxe*, так как размер данной утилиты превышает допустимые нормы прямой загрузки через TFTPД.

```
LABEL HPSmartStart  
kernel pxelinux.cfg/ipxe/ipxe.lkrn  
initrd /LOADER/Utilitys/HPSmartStart.ipxe
```

В следующем блоке находится утилита для диагностики серверов HP, моделей DL180 G6 и DL160 G6. Метод загрузки аналогичен методу загрузки *HPSmartStart*, согласно размеру данной утилиты.

```
LABEL HPEasySetup  
kernel pxelinux.cfg/ipxe/ipxe.lkrn  
initrd /LOADER/Utilitys/HPEasySetup.ipxe
```

В следующем блоке находится утилита для диагностики серверов DELL. Данная утилита имеет не большой размер, исходя из этого, метод загрузки выбран напрямую через TFTPД. Для этого мы подключаем ядро загрузки образов через TFTPД и указываем путь до образа данной утилиты.

```
LABEL DELL Test  
kernel memdisk  
append initrd=Utilitys/DELL.img
```

Аналогичным образом прописан блок для утилиты Victoria. Данная утилита предназначена для диагностики HDD и SSD накопителей.

*LABEL Victoria*

*kernel memdisk iso*

*initrd Utilitys/Victoria.iso*

Так как, утилита Hardware\_Detection\_Tool имеет схожие параметры с предыдущими утилитами, загрузка выбрана напрямую. Данная утилита предназначена для вывода всей информации об аппаратной части сервера.

*LABEL Hardware\_Detection\_Tool*

*kernel memdisk iso*

*initrd Utilitys/Hardware\_Detection\_Tool.iso*

Последняя используемая утилита имеет Linux формат и малый размер поэтому загрузка прописана как подключение ядра, что обеспечивает высокую скорость.

*LABEL MEMTest*

*kernel Utilitys/memtest86*

Типовой блок возврата в основное меню.

*LABEL <- Main Menu*

*KERNEL vesamenu.c32*

*APPEND pxelinux.cfg/default*

Набор данных утилит покрывает все необходимые этапы диагностики и обеспечивает полную проверку всех составляющих сервера.

Исходя из этого, подменю выглядит так, как показано на рисунке 2.14.

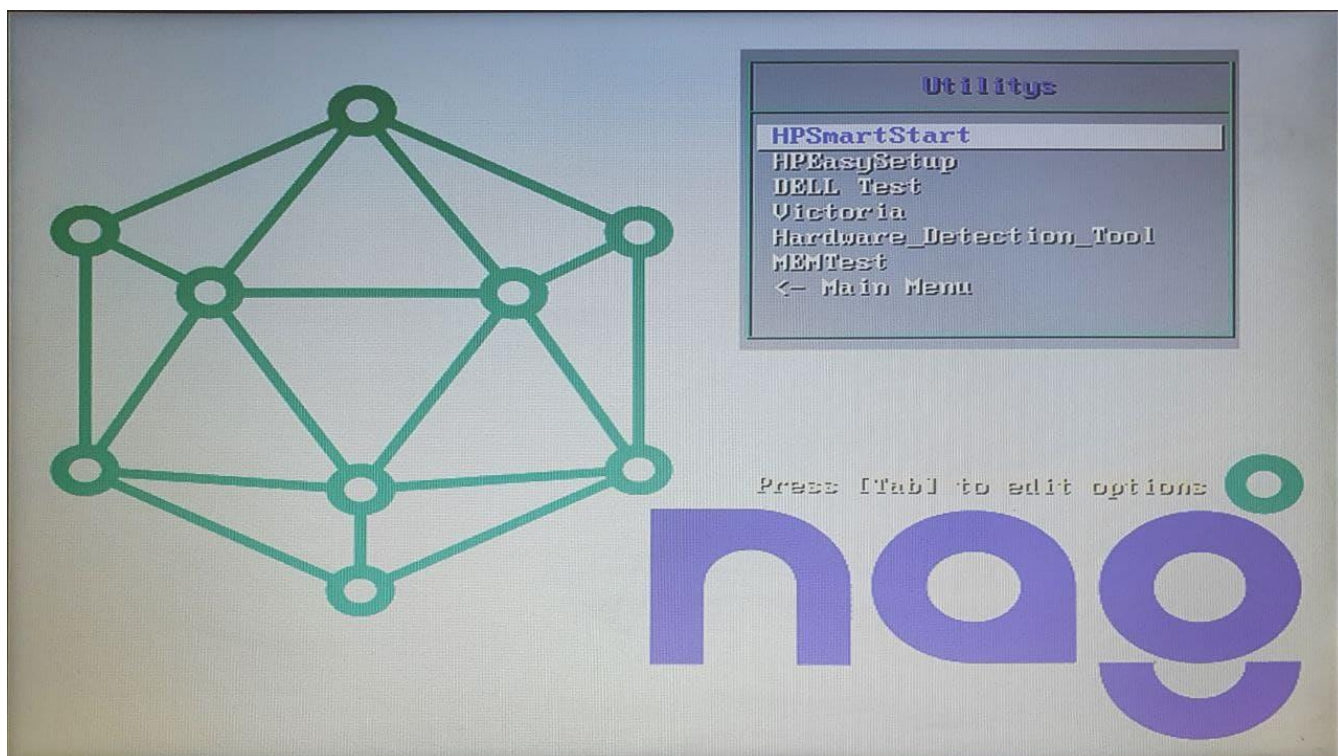


Рисунок 2.14 – Загрузочный экран для диагностических утилит

Конфигурационный файл VMWare.

Данный файл добавлен для установки системы VMWare. Так как, данный PXE сервер предназначен для отдела диагностики серверов и СХД, часто требуется установить на клиентские машины систему виртуализации. Структура файла аналогична файлам с загрузкой через ipxe.

```
default vesamenu.c32
PROMPT 0
MENU INCLUDE /pxelinux.cfg/graphics
MENU TITLE Debian
LABEL VMWare 6 install
kernel pxelinux.cfg/ipxe/ipxe.lkrn
initrd /LOADER/InstallOS/VMWare.ipxe
LABEL <- Main Menu
KERNEL vesamenu.c32
APPEND pxelinux.cfg/default
```

При необходимости можно добавить другие версии VMWare, но так как обычно требуется последняя версия данного ПО, было решено добавить только шестую версию.

Исходя из этого, подменю выглядит так, как показано на рисунке 2.15.

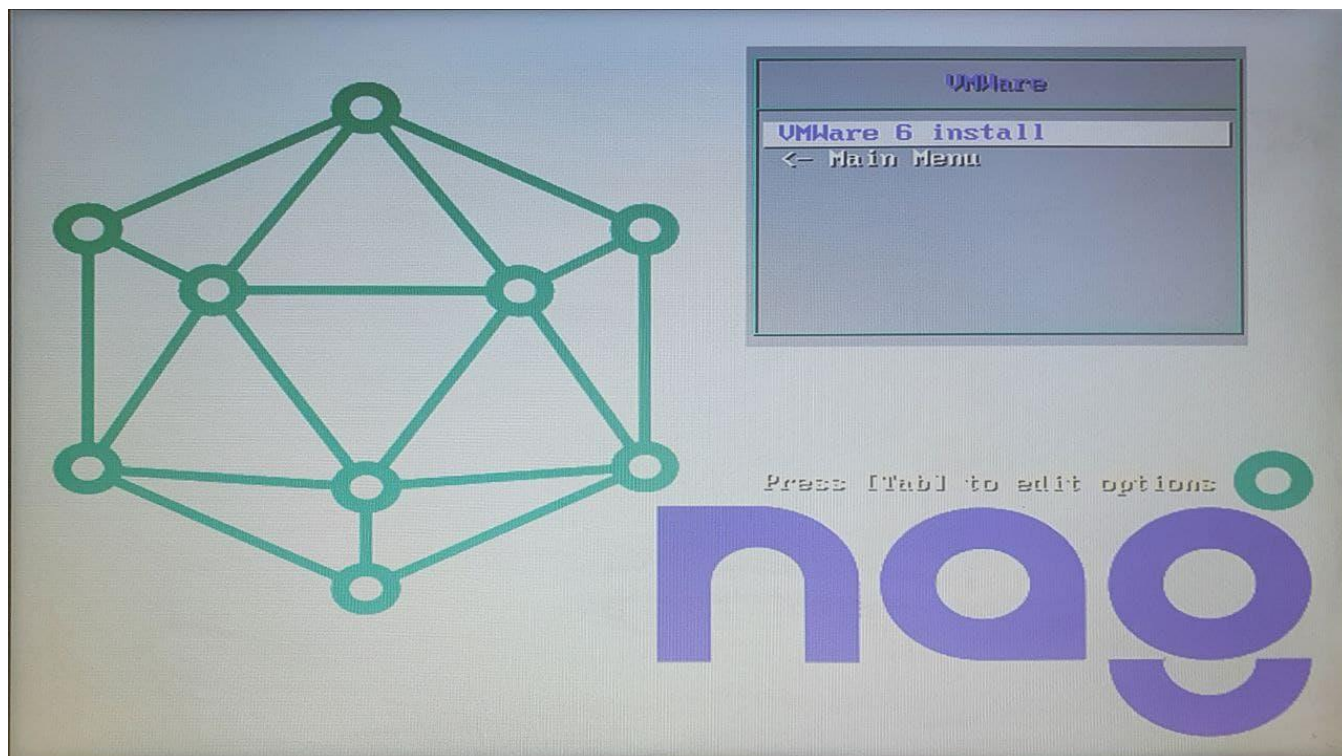


Рисунок 2.15 – Загрузочный экран для установки VMWare 6

Подробно рассмотрев конфигурацию каждого файла меню, как основного, так и второстепенного, перейдем к разбору файла `ipxe` и для чего он требуется.

Разбор структуры файла `ipxe` и описание потребности в нем.

Так как, данный конфигурационный файл требуется для загрузки объемных образов программного обеспечения, он используется для многих образов в данной конфигурации PXE-сервера, но для каждого образа структура файла однотипна. Поэтому разберем его на примере одного из них.

Первым делом прописываем запуск оболочки `ipxe`.

```
#!/ipxe
```

Далее прописываем команду, которая активирует получение реквизитов по DHCP.

```
dhcp
```



Так как этот файл является скриптом, нужно прописать выбор первой ссылки на загрузку, иначе действие прервётся и загрузка образа не будет выполнена.

```
set keep-san 1
```

Подключаем сетевое хранилище и прописываем ссылку на наш Web-сервер, до нужного нам образа.

```
sanboot --no-describe -k http://10.10.10.14/LiveOS/WinPE10.iso
```

После данных конфигураций загрузка наших образов будет происходить через Web-сервер.

На данном этапе настройка PXE сервера и все сопутствующих компонентов закончена. Далее разберем настройку сервера для доступа через SSH по удаленному каналу.

Удаленное управление сервером PXE.

Данное решение предназначено для управления, мониторинга и обслуживания сервера PXE. Поэтому настроим модуль wifi данного сервера наподобие консольного порта.

Для этого нам потребуется выполнить следующие действия:

Устанавливаем модуль для преобразования модуля wifi в точку доступа.

```
apt-get install dnsmasq dkms hostapd -y
```

После этого, переходим в конфигурационный файл сетевых интерфейсов и дополняем его и настройками сетевых реквизитов для wifi интерфейса.

```
nano /etc/network/interfaces
```

Вводим уже знакомые нам параметры, но для интерфейса wifi.

```
auto wlan0
```

```
iface wlan0 inet static
```

```
address 172.24.1.1
```

```
netmask 255.255.255.0
```

```
network 172.24.1.0
```

По данным сетевым реквизитам будет производиться подключение по SSH.

Открываем конфигурационный файл “*nano /etc/create\_ap.conf*” точки доступа и меняем следующие строки под наши параметры:

Задаем IP адрес сетевого интерфейса WiFi.

*GATEWAY=172.24.1.1*

Задаем сетевой интерфейс для точки доступа.

*WIFI\_IFACE=wlan0*

Задаем имя точки доступа.

*SSID=Armbian*

Задаем пароль для доступа к данной точке.

*PASSPHRASE=1234*

Перезапускаем нашу станцию для корректного применения всех параметров и переходим к диагностике работоспособности.

## 2.2 Описание продукта и демонстрация

Первым делом проверяем работоспособность системы удаленного доступа.

Подключаемся к точке доступа “Armbian” с компьютера как показано на рисунке 2.16.

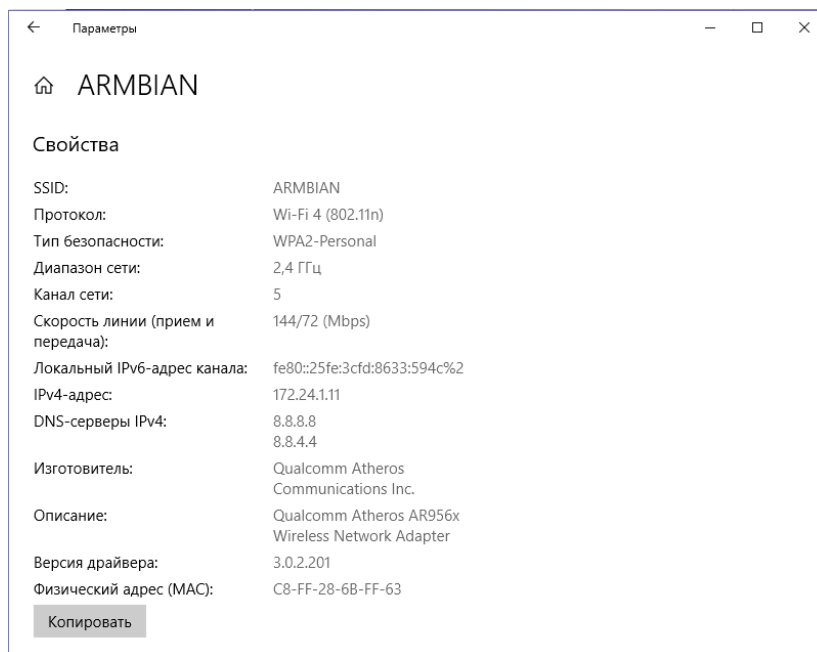


Рисунок 2.16 – Демонстрация подключения к созданной нами точке доступа.

После подключения к точке доступа открываем любой консольный терминал, вводим сетевые реквизиты и выбираем тип подключения по SSH. Далее вводим логин и пароль от системы armbian, как показано на рисунках 2.17.

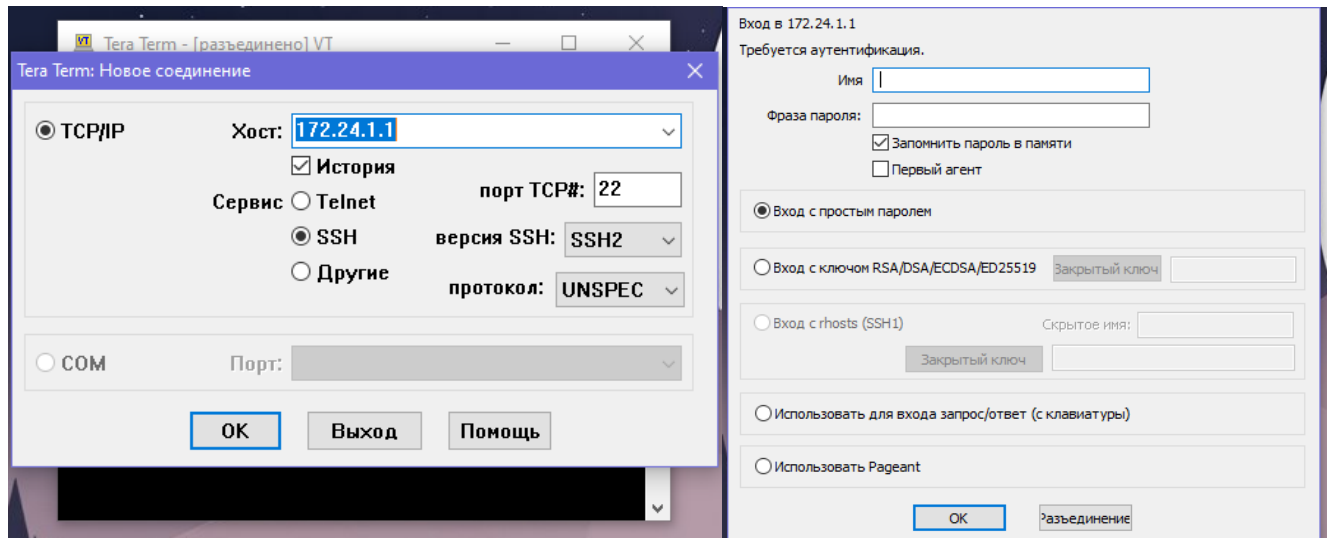


Рисунок 2.17 – Подключение по SSH

В следствии проделанной нами работы, по приготовлению сервера для удаленного подключения к нему инженеров, получаем следующий результат, показанный на рисунке 2.18

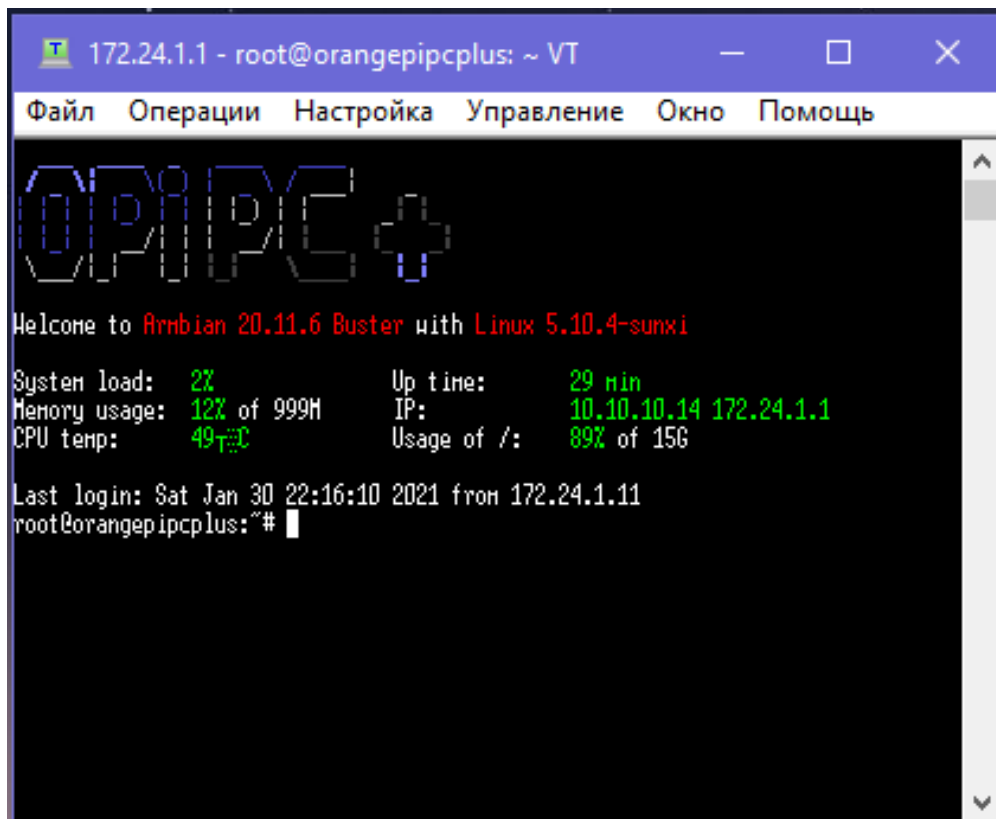


Рисунок 2.18 – Демонстрация удаленного подключения.



На рисунке 2.18, где демонстрируется возможность удаленного подключения к серверу, мы можем наблюдать информацию о состоянии данного сервера, а также реквизиты доступа. Таким образом можно осуществлять мониторинг состояния данной системы.

Кроме мониторинга состояния, мы можем осуществлять настройку системы по средствам CLI. Все манипуляции, которые мы проводили с сервером от первого включения до данного момента, можно осуществить через данное подключение.

Подключение к данному серверу можно осуществлять не только через компьютер, другой сервер или ноутбук, но и через мобильное устройство. Так как подключение беспроводное, мы имеем возможность подключиться к нему через телефон, что предоставляет нам удобство обслуживания данной системы находясь не на рабочем месте.

На рисунках 2.19 – 2.20 продемонстрирован способ подключения через мобильное устройство.

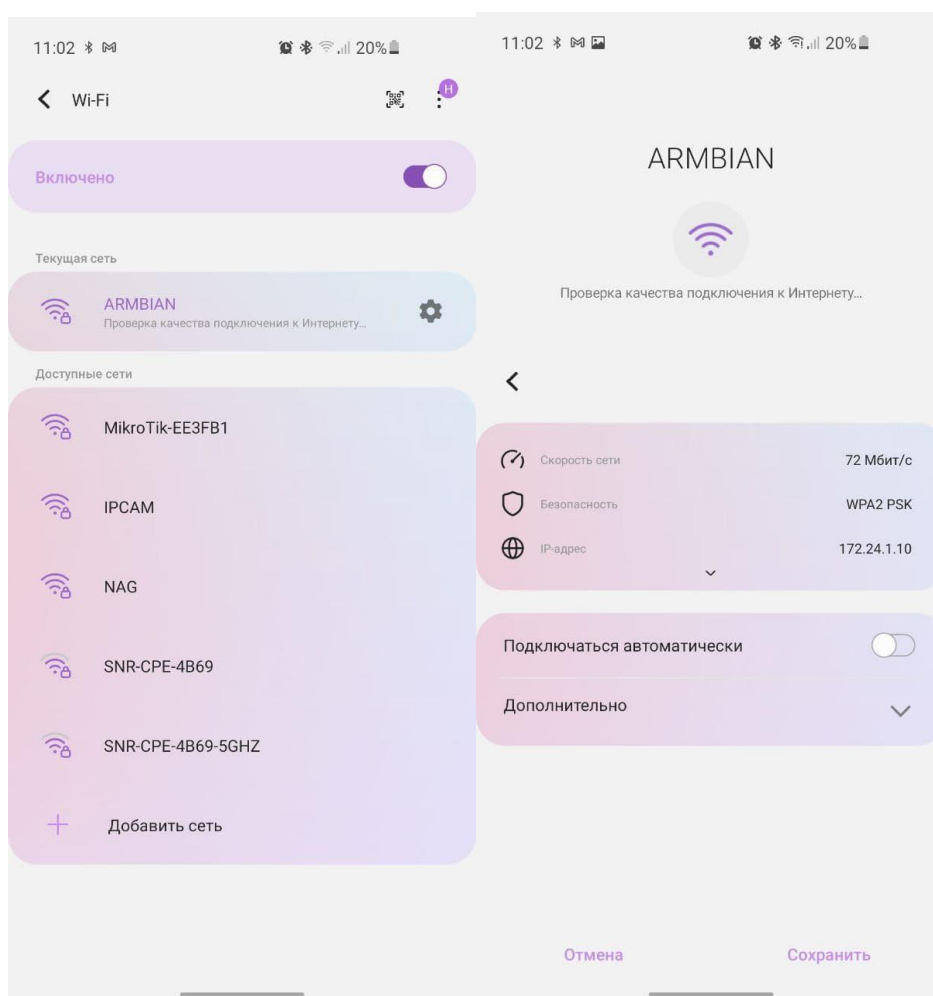


Рисунок 2.19 – Подключение к точке доступа через телефон

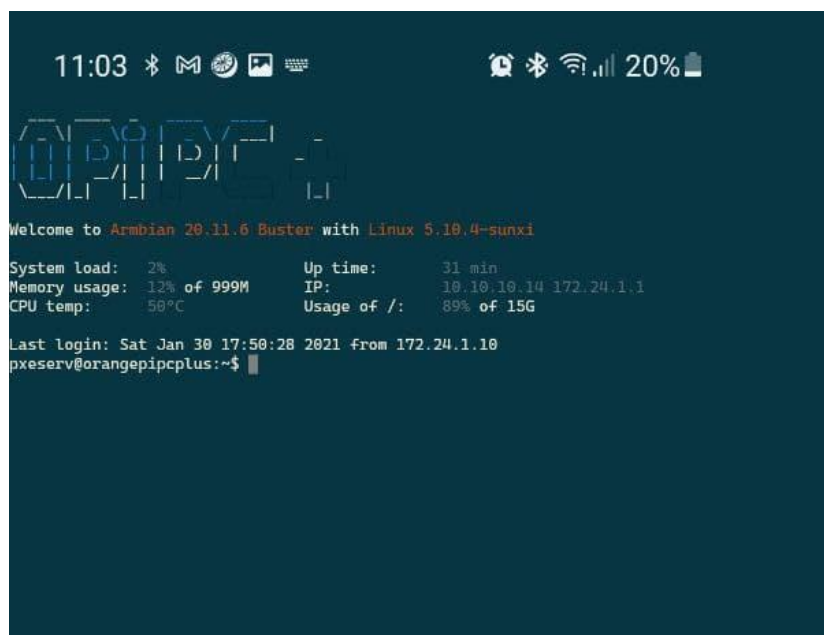


Рисунок 2.20 – Демонстрация подключения к серверу через телефон

Работа системы через удаленный доступ продиагностирована и продемонстрирована. Переходим к демонстрации работоспособности серверной части.

Демонстрация работы серверной части.

Подключив созданный сервер к рабочей станции пользователя по средствам Ethernet кабеля, подключение продемонстрировано на рисунке 2.21, при загрузке клиентской машины она автоматически получает IP-адрес из определенного ранее пула адресов.

Получив реквизиты сети, происходит синхронизация с сервером TFTPД, который автоматически загружает файл `pxelinux.0`, который отвечает за выгрузку всех остальных файлов из созданных директорий.

Процесс данной работы показан на рисунке 2.22.



Рисунок 2.21 – Способ подключения сервера к рабочим станциям

Для подключения сервера к рабочим станциям используется коммутатор.

Данное решение дает возможность одновременного подключения нескольких рабочих станций для их связи с PXE-сервером.

```
Intel(R) Boot Agent GE v1.5.53
Copyright (C) 1997-2014, Intel Corporation

CLIENT MAC ADDR: F8 F0 82 35 06 70  GUID: 00020003 0004 0005 0006 000700080009
CLIENT IP: 10.10.10.101  MASK: 255.255.255.0  DHCP IP: 10.10.10.14
GATEWAY IP: 10.10.10.1

PXELINUX 4.05 2011-12-09  Copyright (C) 1994-2011 H. Peter Anvin et al
!PXE entry point found (we hope) at 95F2:0106 via plan A
UNDI code segment at 95F2 len 5DB0
UNDI data segment at 8FB2 len 6400
Getting cached packet  01 02 03
My IP address seems to be C0A80148 10.10.10.101
ip=10.10.10.101: 10.10.10.14:10.10.10.1:255.255.255.0
BOOTIF=01-f8-f0-82-35-06-70
SYSUUID=00020003-0004-0005-0006-000700080009
TFTP prefix:
Trying to load: pxelinux.cfg
```

Рисунок 2.22 – Получение реквизитов

На рисунке 2.22 мы можем наблюдать адреса, которые мы настроили ранее, что означает, работоспособность службы DHCP, а также службы TFTP, так как после выдачи реквизитов идет погрузка нашей папки с конфигурационными файлами, а после этого идет загрузка графической части.

На рисунках 2.23 – 2.30 продемонстрирована работа с несколькими рабочими станциями одновременно.

На данных рисунках мы можем видеть одновременную, не зависимую между собой работу на рабочих станциях. На каждой рабочей станции открыто разное меню, а далее происходит загрузка разных программ для демонстрации одновременной работы.

Таким образом подключившись к серверу можно работать и загружать разное программное обеспечение без ущерба в работе другого инженера



Рисунок 2.23 – Одновременная загрузка



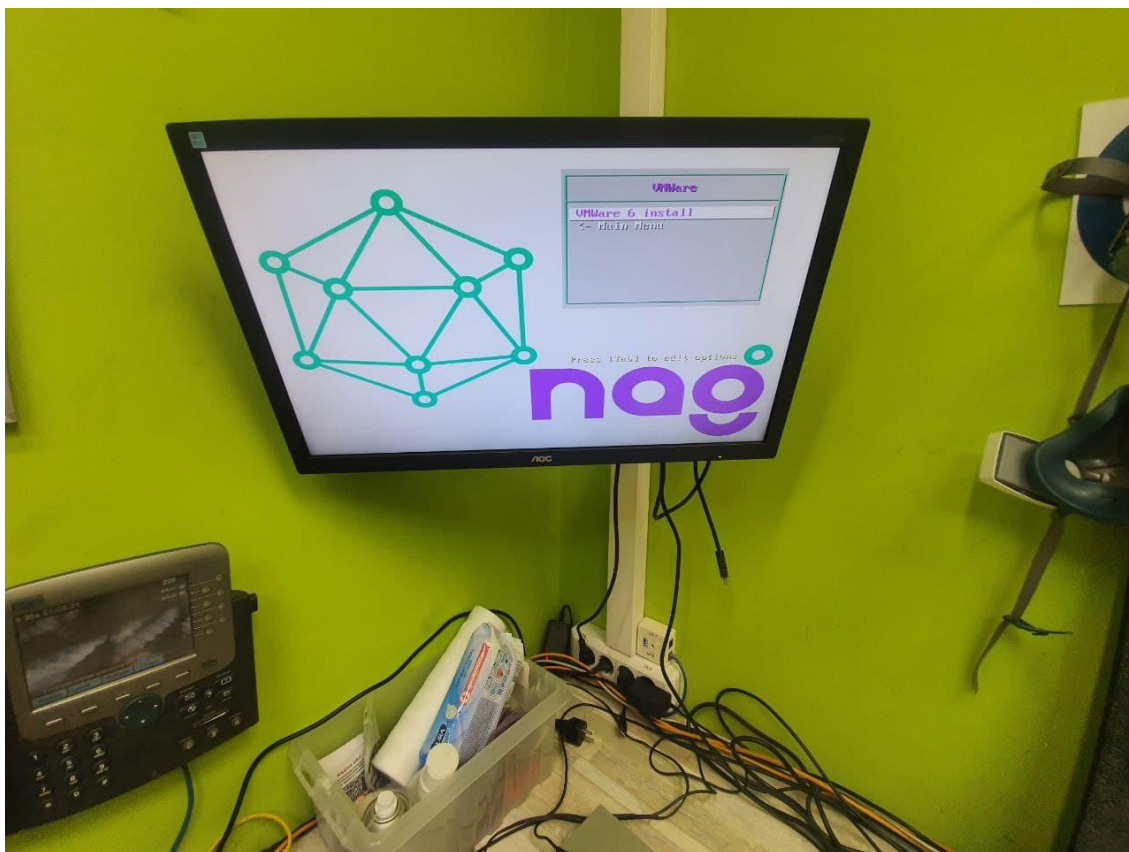


Рисунок 2.24 – Демонстрация экрана первой рабочей станции

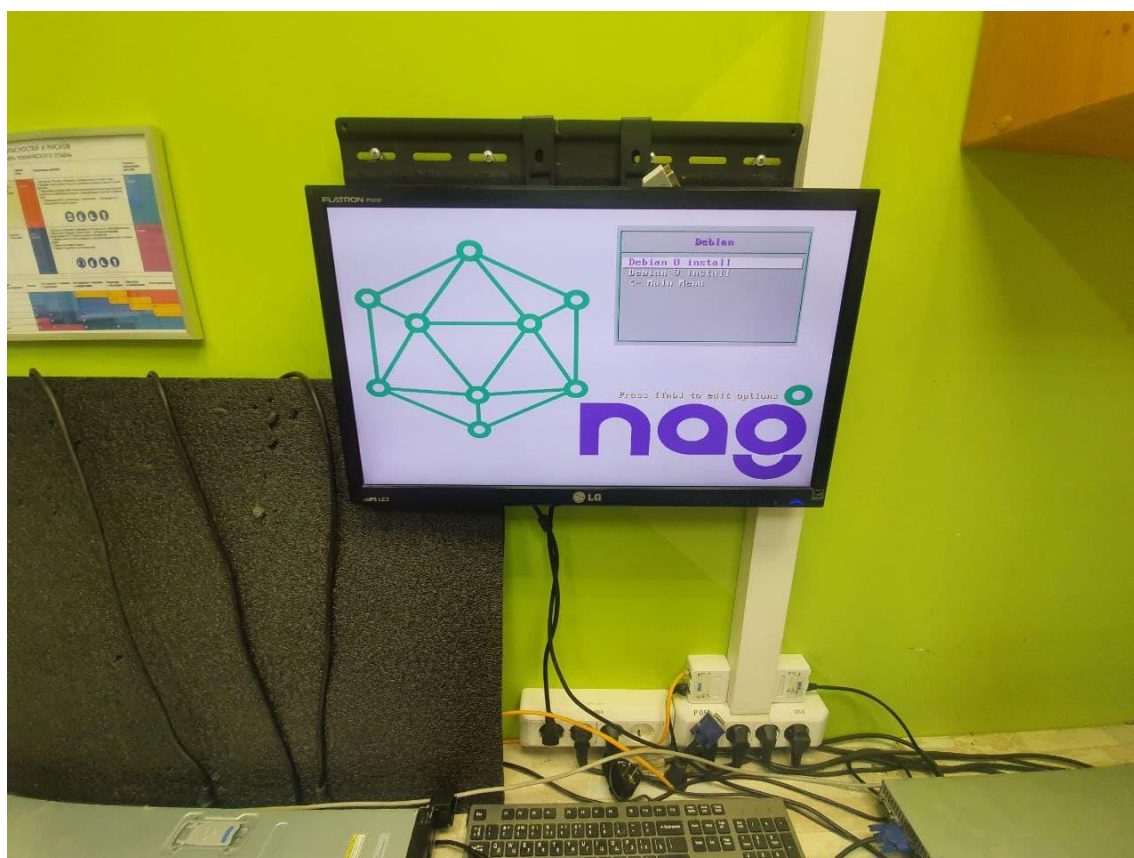


Рисунок 2.25 – Демонстрация экрана второй рабочей станции

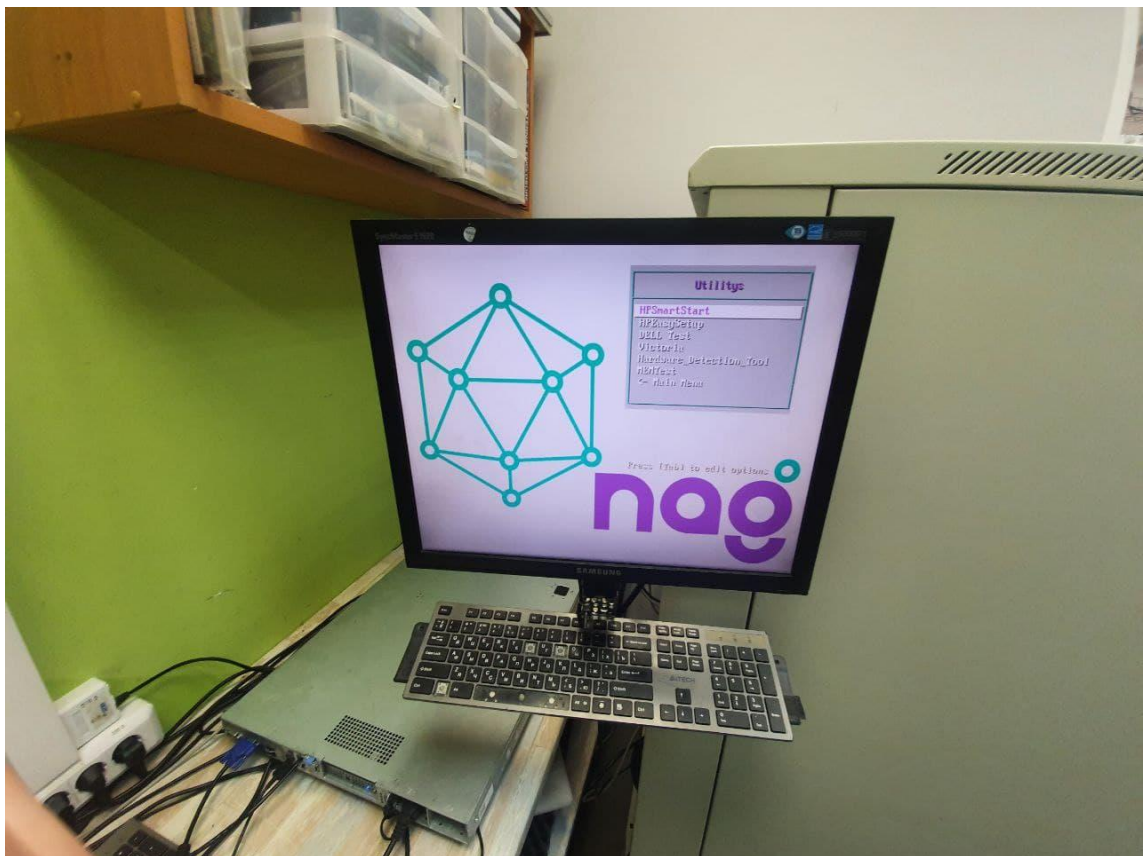


Рисунок 2.26 – Демонстрация экрана третьей рабочей станции



Рисунок 2.27 – Демонстрация загрузки разного ПО одновременно





Рисунок 2.28 – Демонстрация загруженного ПО первой рабочей станции

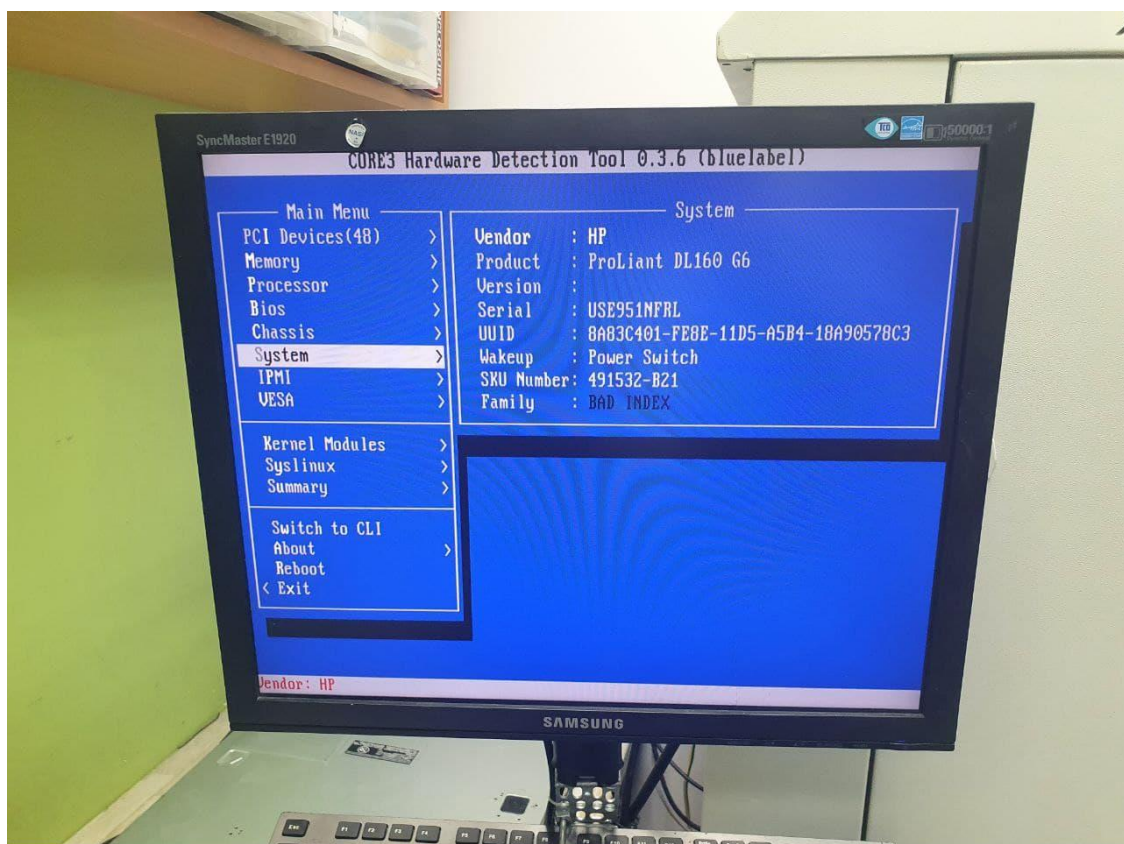


Рисунок 2.29 – Демонстрация загруженного ПО второй рабочей станции

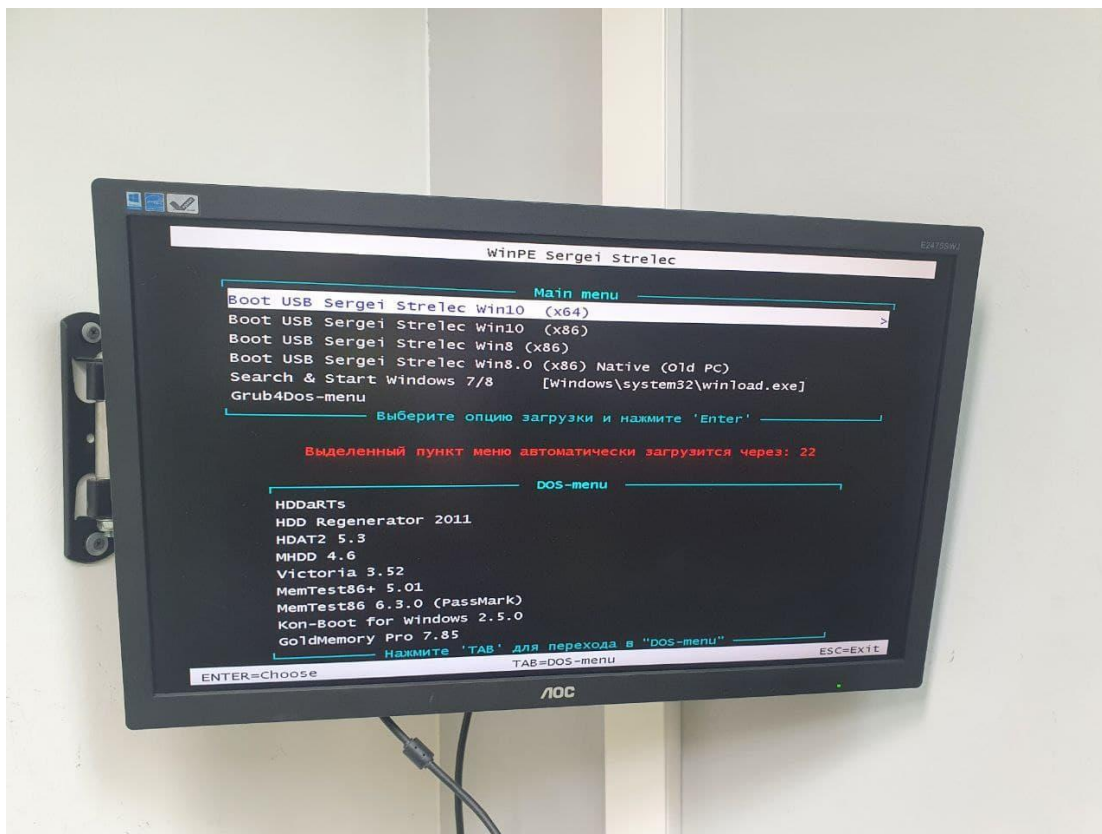


Рисунок 2.30 – Демонстрация загруженного ПО третьей рабочей станции

Также нужно проверить и продемонстрировать работоспособность Web-сервера, с которого происходит загрузка массивных объектов PXE сервера.

На рисунках 2.32 – 2.34 можно наблюдать работоспособность web-сервера, а также доступ к папкам по всем путям в которых лежат отсортированные файлы программного обеспечения.

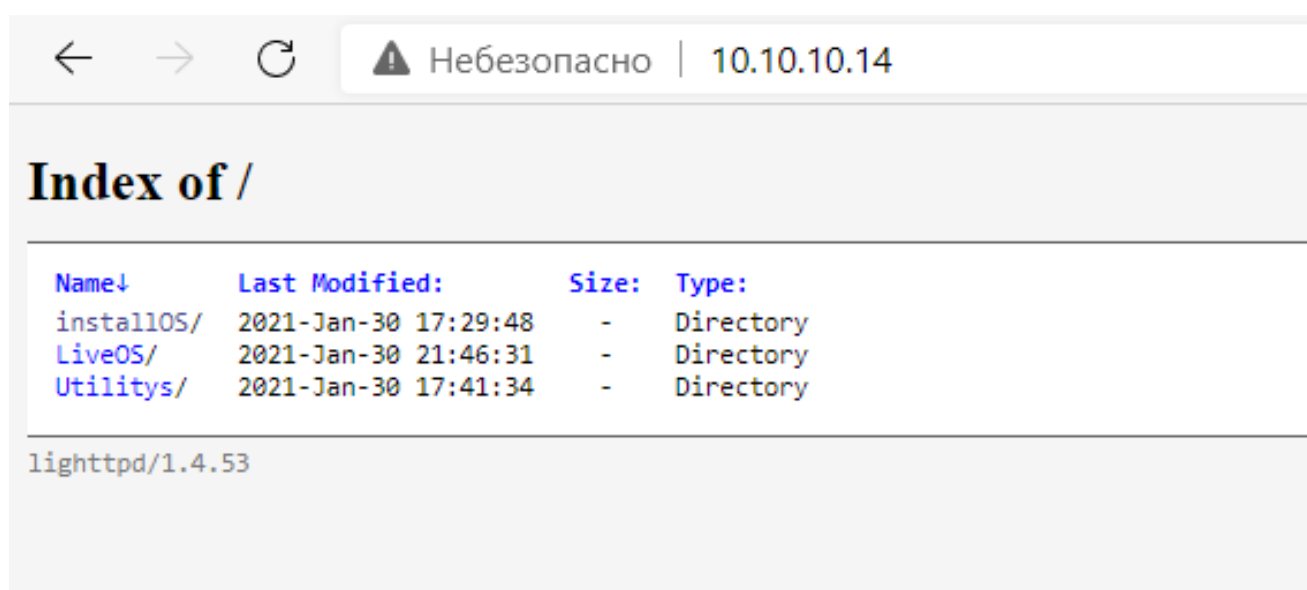


Рисунок 2.31 – Главная страница с основными каталогами



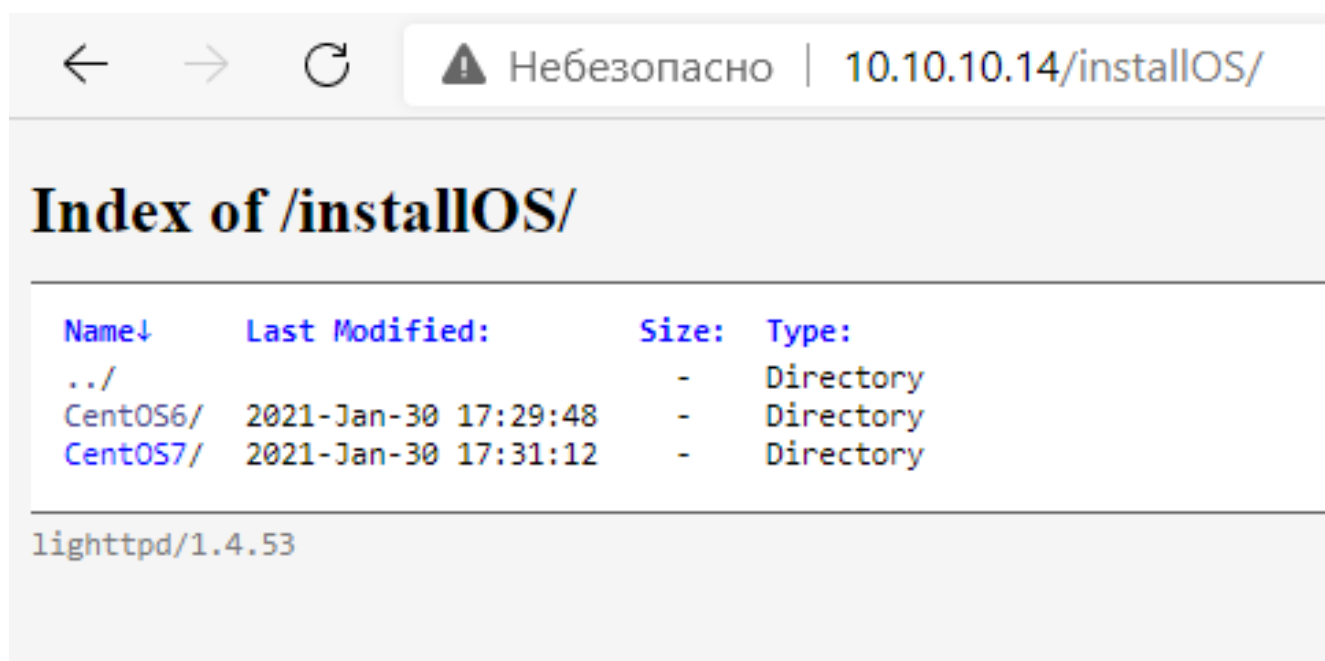


Рисунок 2.32 – Каталог с установочными директориями CentOS

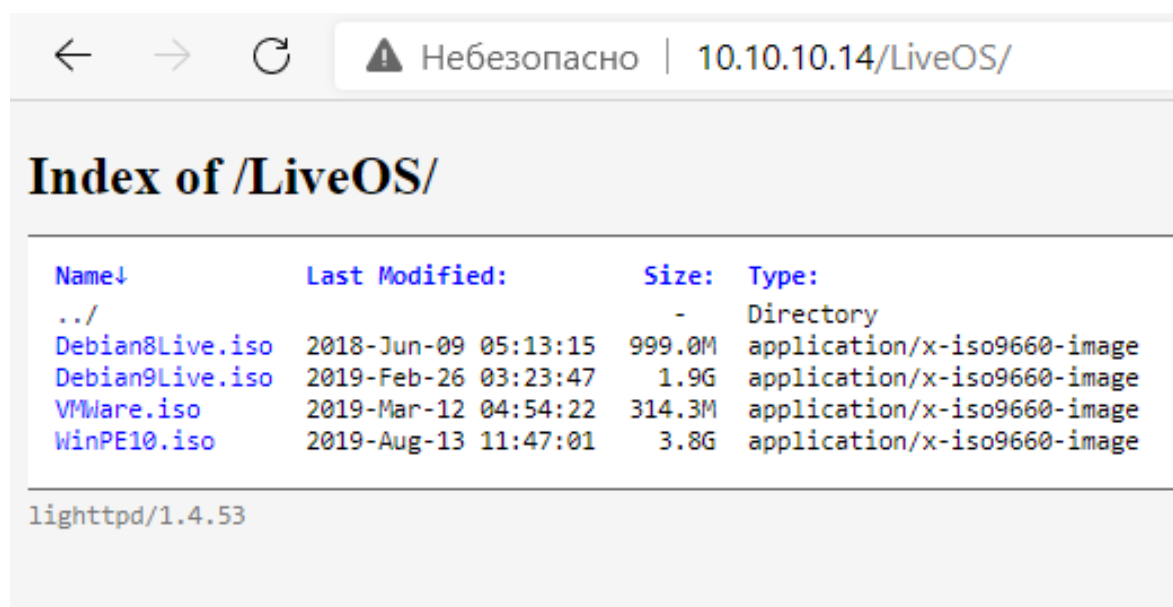


Рисунок 2.33 – Каталог с ISO образами LiveOS

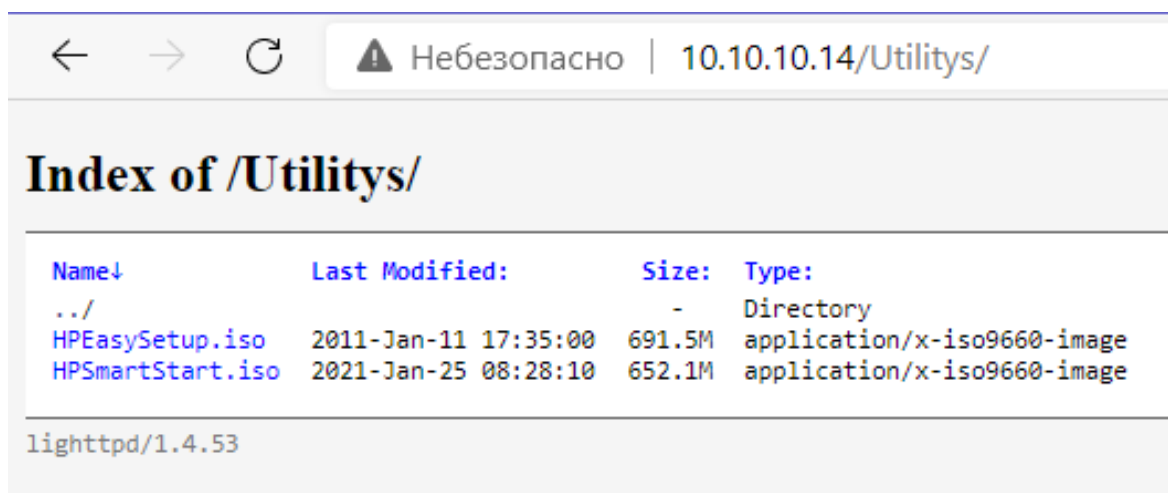


Рисунок 2.34 – Каталог с ISO образами диагностических утилит HP.

Нами была осуществлена проверка работоспособности сервера с несколькими рабочими станциями. Была проверена загрузка разного программного обеспечения и отрисовка каждого системного меню. Также была проверена работоспособность Web-сервера и загрузка программного обеспечения с него. Помимо работоспособности самого сервера, была успешно проверена и продемонстрированная работа удаленного подключения к серверу.

На данном этапе демонстрация всех систем сервера окончена. Все системы работоспособны, в следствии этого демонстрацию можно считать успешной.

## 2.3 Результаты апробации и внедрения.

В ходе первых испытаний было решено использовать данный сервер в специализированном диагностическом телекоммуникационном шкафу, предназначенном для диагностики серверов и серверных компонентов, который продемонстрирован на рисунке 2.35



Рисунок 2.35 – Диагностический телекоммуникационный шкаф с сервером PXE

Сервер PXE вмонтирован в данный шкаф и будет использоваться непосредственно в данном рабочем стенде для проведения лабораторных работ и предпродажных подготовок инженерами технического отдела компании ООО НАГ.

В результате использования данного сервера нареканий не выявлено.

При использовании данной системы отпала необходимость в большом количестве Flash-носителей.

Переход на данную системы оказал положительный эффект для работы инженеров.

Внедрение данной системы в бизнес процессы сократило время на проведение тестирования оборудования.

Бизнес процессы были оптимизированы с внедрением сервера PXE.

Акт апробации и внедрения сервера PXE находится в приложении 1.

## **Заключение**

В ходе данной дипломной работы был разработан сервер PXE с удаленным доступом для технического отдела компании ООО НАГ.

На примерах был показан алгоритм настройки службы и ее практическое применение в сети TCP/IP.

Для загрузки рабочей станции через PXE были организованы взаимодействия служб DHCP, HTTP и TFTP с сервером PXE.

Проделана работа по настройке параметров для функционирования службы PXE.

Разработаны варианты использования службы PXE инженерами технического отдела для упрощения взаимодействия с рабочими станциями при проведении диагностических работ.

Опираясь на приведенные примеры по работе с PXE, мы можем сделать вывод, что служба PXE значительно упрощает настройку оборудования и дальнейшее администрирование.

Также, данная служба является актуальной и использование PXE сервера значительно облегчает работу не только инженеров технического отдела данной компании, но и любого системного администратора.

## Список информационных источников

- 1) Бабушкин А.Н. Что такое персональный компьютер. – Воронеж: Велби, 2009.
- 2) Бэкон Дж., Харрис Т., Операционные системы. – СПб.: Питер, 2004.
- 3) Гук М. Аппаратные средства локальных сетей. Энциклопедия. – СПб.: Питер, 2012.
- 4) Интернет магазин компании NAG[Электронный ресурс] – Режим доступа: <https://shop.nag.ru>, дата обращения к ресурсу 20.01.2021.
- 5) Кузьменко, Н.Г. Компьютерные сети и сетевые технологии – СПб.: Наука и техника, 2013.
- 6) Лимончелли Т. Тайм-менеджмент для системных администраторов. – СПб.: Символ-Плюс, 2007.
- 7) Мюллер С., Соупер М.Е. и Сосинских Б.А. Модернизация и ремонт серверов. – М.: Вильямс, 2009.
- 8) Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. 4-е издание – СПб: Питер, 2013.
- 9) Официальный сайт компании NAG[Электронный ресурс] – Режим доступа: <https://nag.ru>, дата обращения к ресурсу 20.01.2021.
- 10) Официальный сайт фирмы HP[Электронный ресурс] – Режим доступа: <https://www.hpe.com/ru/ru/home.html>, дата обращения к ресурсу 20.01.2021.
- 11) Поляк-Брагинский А. Локальные сети. Модернизация и поиск неисправностей. – М.: БХВ-Петербург, 2012.
- 12) Прончев Г. Б., Бухтиярова И. Н., Брутов В. В., Фесенко В. В. Компьютерные коммуникации. Простейшие вычислительные сети. – М.: КДУ, 2009.
- 13) Пятибратов А. П., Гудыно Л. П., Кириченко А. А. Вычислительные системы, сети и телекоммуникации. – М.: КноРус, 2013.
- 14) Семенов А.Б. Проектирование и расчет структурированных кабельных систем и его компонентов. – М.: Компания АйТи, 2013.
- 15) Таненбаум Э. С., Уэзеролл Д. Компьютерные сети. 5-е изд. – СПб: Питер, 2012.

- 16) Флинт Д. Локальные сети ПК: принципы построения. – М.: Финансы и статистика, 2012.
- 17) Флинт, Д. Локальные сети ЭВМ: архитектура, принципы построения, реализация. – М.: Финансы и статистика, **2013**.
- 18) Фролов, А.В.; Фролов, Г.В. Локальные сети персональных компьютеров. Использование протоколов IPX, SPX, NETBIOS. – М.: Диалог-Мифи, **2013**.
- 19) Хомоненко А.Д. Основы современных компьютерных технологий. – СПб.: Корона, 2008.
- 20) Корячко В. П., Перепелкин Д. А. Корпоративные сети. Технологии, протоколы, алгоритмы – М.: Горячая линия - Телеком, 2012.

# Приложения

## Приложение 1.

### АКТ АПРОБАЦИИ И ВНЕДРЕНИЯ

РХЕ-сервера с удаленным доступом для технического отдела компании ООО «НАГ»

Составлен «10» сентября 2021 года.

В телекоммуникационной компании ООО «НАГ» г. Екатеринбург в рамках выпускной квалификационной работы «РХЕ-сервер с удаленным доступом» была проведена апробация и внедрение сервера РХЕ.

В процессе апробации были выполнены следующие работы:

1. Выявлена потребность в сервере РХЕ;
2. Создан список необходимого программного обеспечения для загрузки через РХЕ;
3. Выполнена установка и конфигурация сервера РХЕ;
4. Добавлены образы программного обеспечения на сервер РХЕ;
5. Интерфейс разработан в стилистики компании ООО «НАГ»;
6. Настроен удаленный доступ для мониторинга и конфигурации сервера.

**Заключение по результатам апробации и внедрения сервера РХЕ с удаленным доступом для технического отдела компании ООО «НАГ».**

Работы по плану апробации и сервера РХЕ с удаленным доступом проведены в полном объеме.

Сервер РХЕ организации удобен в использовании и содержит все необходимое программное обеспечение для проведения предпродажных подготовок и создания лабораторных стендов.

Руководитель технического отдела компании ООО «НАГ» Алексеев С.А.





Министерство просвещения РФ  
ФГБОУ ВО «Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологий  
Кафедра информатики, информационных технологий и методики обучения информатике

ОТЗЫВ

на выпускную квалификационную работу

Студента **Сливканич Д.Д.** Группа **ПИ-1601z**

Отделение заочное

Тема работы: **Разработка РХЕ сервера с удаленным доступом**

№	Показатели качества работы студента	Уровень (высокий, средний, низкий)
1	Степень выполнения задач исследования	высокий
2	Умение работать с научной литературой	средний
3	Умение самостоятельно осваивать новое ПО	высокий
4	Владение новыми информационными технологиями	высокий
5	Уровень самостоятельности студента	высокий
6	Степень ответственности	высокий
7	Инициативность	высокий
8	Исполнительность	высокий
9	Соблюдение сроков плана выполнения ВКР (систематичность)	средний

Наличие Актов внедрения продукта – есть.

Наличие публикаций – нет.

Работа студента **Сливканич Д.Д.** отвечает требованиям, предъявляемым к выпускным работам, и он заслуживает присвоения квалификации «бакалавр» по направлению подготовки 09.03.03 Прикладная информатика.

Научный руководитель:

кандидат педагогических наук, доцент Рожина Ирина Веникентьевна

Должность:

доцент кафедры ИИТиМОИ

Дата 22 февраля 2020 г.

Подпись \_\_\_\_\_ (Рожина И.В.)